Good Regions to Deblur

Zhe Hu and Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
{zhu,mhyang}@ucmerced.edu

Abstract. The goal of single image deblurring is to recover both a latent clear image and an underlying blur kernel from one input blurred image. Recent works focus on exploiting natural image priors or additional image observations for deblurring, but pay less attention to the influence of image structures on estimating blur kernels. What is the useful image structure and how can one select good regions for deblurring? We formulate the problem of learning good regions for deblurring within the Conditional Random Field framework. To better compare blur kernels, we develop an effective similarity metric for labeling training samples. The learned model is able to predict good regions from an input blurred image for deblurring without user guidance. Qualitative and quantitative evaluations demonstrate that good regions can be selected by the proposed algorithms for effective image deblurring.

1 Introduction

Motion blur on an image often results from the relative motion between a camera viewpoint and the scene (e.g., camera shake) at the exposure time. It causes significant image degradation, especially in the low light conditions where longer exposure time is required. Recovering the latent image from one single blurred image has been studied extensively with a rich literature. Typically, the blurred image formation process is modeled as a latent image convolved with a spatial-invariant blur kernel (i.e., the point spread function). Hence, the deblurring process is known as a 2D deconvolution problem. When the underlying blur kernel is known or has been accurately estimated, the problem is reduced to non-blind deconvolution. On the other hand, if the blur kernel is unknown, the deblurring problem is known as blind deconvolution. The ill-posed nature of the single image deblurring setting makes the problem rather difficult.

To deblur an image, it is shown that estimating the blur kernel first and then solving a non-blind deconvolution problem with the estimated kernel renders favorable results [1]. In this case, the performance of the blur kernel estimation directly affects the performance of the deblurred results (i.e., the estimated latent image). That is, one can recover the latent image well if the blur kernel can be accurately estimated.

For the single image deblurring problem, it is usually advantageous to make full use of the input blurred image. However, not all pixels of the input blurred image are informative. Smooth regions, for example, do not contribute much for

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part V, LNCS 7576, pp. 59–72, 2012.

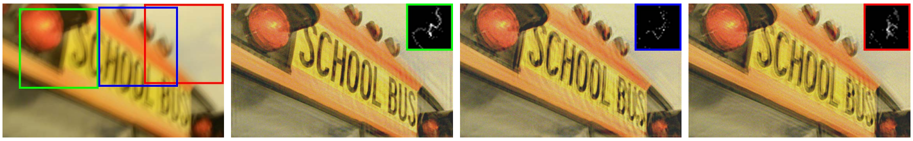


Fig. 1. Different regions lead to different kernel estimations and deblurred results. The top left image is the input blurred image with three subwindows selected for estimating kernels. The other three images are the recovered images and estimated kernels from these three subwindows using [2].

estimating the blur kernel. In this paper, we ask "What kind of image features or structures of a blurred image really help in kernel estimation?" By detecting good image features for blur kernel estimation, an accurately estimated blur kernel can then be used to recover a latent clear image with high visual quality. In [1, 2], it is demonstrated that regions with strong edges tend to yield better deblurring results. Some of the gradient-based methods favor salient edges with gradients of specific patterns [3–5]. On the other hand, based on 1D signal examples, it is demonstrated that edges of short length could adversely affect the deblurring results [5]. In other words, if the whole image is used for image deblurring without deliberate selection of good features, negative impacts are likely to lead to inferior results. For this reason and the computational efficiency issue, it is preferable to determine a region, rather than the whole image, for estimating blur kernels. Figure 1 illustrates that different regions may lead to completely different kernel estimation results, and thereby different recovered images. This problem can often be partly alleviated by manual selection and visual inspection of the results. However, this requires tedious human input for deblurring images. In addition, the questions regarding which regions or what image structures are crucial for accurate blur kernel estimation remain unanswered.

In this paper, we address these questions for effective and efficient image deblurring. We first propose a metric that quantitatively measures the similarity between kernels, which facilitates the process of labeling good estimated kernels. Instead of determining good image structures from empirical understanding and prior knowledge, we resort to learning for this task based on a collection of labeled data with the proposed kernel similarity measure. We pose the learning problem within the Conditional Random Field (CRF) [6] framework in order to exploit contextual constraints among image regions. In addition, we explore the contribution of different features with structured output. We construct a dataset which covers large variability of image structure and blur kernel following the technique described in [1] for evaluation, and apply the learned models to select good image regions for deblurring. Experimental results demonstrate the effectiveness and efficiency of our algorithms for selecting good regions to deblur.

2 Related Work

Image deblurring has been studied extensively and numerous algorithms have been proposed. Here we briefly discuss the most related algorithms and put this work in proper context. Since blind deconvolution is an ill-posed problem, prior knowledge or additional information are often required for effective solutions. In the image deblurring literature, two types of additional information are often used: natural image priors and additional image observations.

One line of research focuses on exploring image priors for deblurring. In [2], the heavy-tailed gradient distribution of natural images are exploited as prior information. The mixture of Gaussian approximation is constrained to fit the distribution of gradient magnitudes of natural images. The sparse gradient prior is also used to search for blur kernels in [7]. In [8], a method is presented to exploit the underlying relation between the motion blur and blurry object boundary, which is shown to facilitate better kernel estimation. In [9], a deblurring algorithm is proposed in which prior knowledge regarding gradients of natural images is used with additional constraints on consistence of local smooth region before and after blurring. The consistency constraints are shown to be effective in suppressing ringing effects. A method that uses sparsity constraints for both blur kernel and latent image in the wavelet domain is presented in [10]. In contrast to existing works that exploit heavy-tailed gradient distribution of natural images, an image restoration algorithm that applies adaptive priors based on texture contents is proposed in [4]. Experimental results on denoising and deblurring show that adaptive priors are important for deblurring results when patches at different image locations are manually selected.

Another line of research tackles image deblurring by leveraging additional image observations. With both low-resolution video camera and high-resolution digital camera, an algorithm that utilizes both spatial and temporal information is proposed [11] for effective image deblurring. On the other hand, noisy images also provide useful information for image deblurring. When a pair of blurred and noisy images of the same scene are available, it has been shown that blur kernel can be estimated using the sharp image structures in the noisy image [12].

Numerous studies focus on exploiting additional information to facilitate image deblurring. Considerably less attention has been paid to exploit image structure for kernel estimation and deblurring. In this paper, we aim to determine useful image structures for kernel estimation and image deblurring.

3 Kernel Similarity

Existing methods mostly resort to visual quality of deblurred images for empirical evaluation. While it is important to recover high visual quality images, it is neither reliable nor effective to evaluate recovered results visually since human vision is sensitive to noise and ineffective in telling minute difference. As suggested in [1], it is preferable to separate the image deblurring problem into two steps: 1) blur kernel estimation and 2) non-blind deconvolution. If the blur kernel can be accurately estimated, then the deblurred image can be easily recovered with non-blind deconvolution algorithms. Therefore, the ensuing question is how to identify kernels effectively.

The difficulty of comparing kernels arises when kernels vary in terms of shift and scale (i.e., the size of the kernel). Two kernels K_1 and K_2 are considered

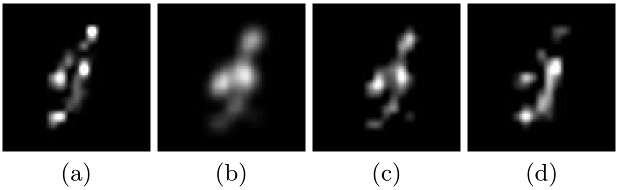


Fig. 2. RMSE and kernel similarity (KS). (a) the ground truth blur kernel; (b) the smooth estimated kernel, with RMSE 0.0013 and KS 0.8178 compared with (a); (c) the estimated kernel has an extra part, with RMSE 0.0016 and KS 0.8791; (d) the estimated kernel misses some parts, with RMSE 0.0012 and KS 0.8106. While kernels in (b) and (d) have smaller RMSEs, the kernel in (c), with higher KS, is closest to the one in (a).

shift and scale invariant if the dominant (i.e., non-zero) parts of them are the same, regardless of differences in locations and sizes of the kernel windows. Thus, a good metric for kernel similarity should be shift and scale invariant.

The commonly used root-mean-square-error (RMSE) metric is not effective in computing the similarity between two kernels. Typically, smooth kernels are favored by RMSE, due to its 2-norm form and the fact that the entries of blur kernel sum up to one. To deal with the above-mentioned problems, we propose a kernel similarity metric to effectively compare estimated kernels with the ground truth. We utilize the maximum response of normalized cross-correlation to represent the blur kernel similarity $S(K,\hat{K})$ of two kernels, K and $\hat{K}$,

$$S(K,\hat{K}) = \max_{\gamma} \rho(K,\hat{K},\gamma), \tag{1}$$

where $\rho(\cdot)$ is the normalized cross-correlation function and γ is the possible shift between the two kernels. Let τ represent element coordinates, $\rho(\cdot)$ is given by

$$\rho(K,\hat{K},\gamma) = \frac{\sum_{\tau} K(\tau) \cdot \hat{K}(\tau+\gamma)}{\parallel K \parallel \cdot \parallel \hat{K} \parallel}, \tag{2}$$

where $\parallel \cdot \parallel$ is 2-norm, $K(\tau)$ and $\hat{K}(\tau)$ are zeros when τ is out of the kernel range. The maximum response of the normalized cross-correlation, similar in nature to the convolution, can handle the shift and scale problems mentioned above. With this metric, the larger kernel similarity values reflect more accurate kernel estimation results. One example that demonstrates the effectiveness of the kernel similarity is shown in Figure 2. We note that the estimated kernels with low RMSEs do not necessarily lead to good results, while estimated kernels with slightly higher RMSEs in fact match better with ground truth kernels. Instead, the kernel similarity is a better metric for comparing kernels. More results can be found in the supplementary material.

4 Learning Good Regions

Existing deblurring works discuss some potential features that may be useful for kernel estimation mainly based on empirical experimental results. In this paper, we address this problem by learning good image regions for deblurring.

4.1 Learning Framework

To determine the good image regions for deblurring, we analyze the image structure by small subwindows. The subwindows within the image, in the context of kernel estimation and recovered image, are spatially dependent. Two closely overlapping subwindows (e.g., shifted by a few pixels in either directions) share similar image structures. In addition, it is reasonable to expect that other subwindows, nearby a potential good subwindow for kernel estimation, contain useful image structures for deblurring. Consequently, the deblurred results for these subwindows should be similar, which are also observed empirically in image deblurring results. Figure 3 shows one example where we estimate a blur kernel from each subwindow of size 200×200 and apply it to recover the whole image. With all the estimated kernels from subwindows and recovered images, we construct an image reconstruction error map (Figure 3(b)) and a kernel similarity map (Figure 3(c)) by comparing with ground truth clear image and blur kernel. The value at each pixel of these maps is computed by averaging the reconstruction errors or kernel similarity values from all the subwindows containing it. The image reconstruction map illustrates that deblurred results using subwindows for kernel estimation are spatially correlated. In the meanwhile, high kernel similarity values well match low image reconstruction errors which demonstrates the effectiveness of kernel similarity as a metric for evaluating deblurred results. Thus, we pose the problem of learning good regions within the CRF framework [6] as it encourages spatial correlation and label the training data using kernel similarity.

Let S and i represent the set of nodes and node index. Given the labels $\mathbf{y}$ and the observations $\mathbf{x}$, the conditional distribution $P(\mathbf{y}|\mathbf{x})$ is

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp\left(E(\mathbf{y}|\mathbf{x})\right), \tag{3}$$

where Z is a normalization term also known as the partition function. The energy E is

$$E(\mathbf{y}|\mathbf{x}) = \sum_{i \in S} A_i(y_i, \mathbf{x}) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I_{ij}(y_i, y_j, \mathbf{x}), \tag{4}$$

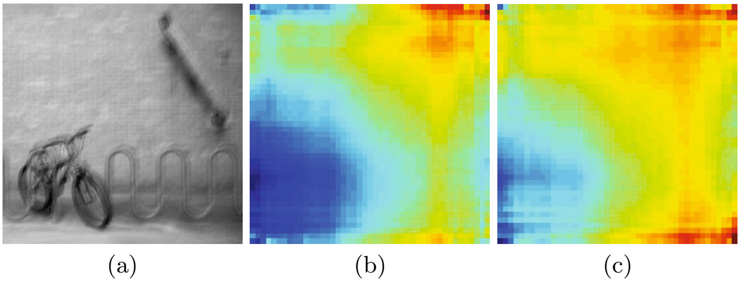


Fig. 3. Spatial correlation. (a) input blurry image; (b) image reconstruction error map built upon the estimated kernels from shifting subwindows (blue to red pixels indicate low to high reconstruction errors compared with the ground truth image); (c) kernel similarity map (blue to red pixels indicate high to low kernel similarity compared with the ground truth kernel).

where A_i and I_{ij} denote the association and interaction potentials, respectively. The association potential $A_i(y_i, \mathbf{x})$ measures how likely the node of index i would be labeled y_i given the observation $\mathbf{x}$. Meanwhile, the interaction potential $I_{ij}(y_i, y_j, \mathbf{x})$ determines how the label y_j at node j affects the one y_i at node i. Here $\mathcal{N}_i$ represents the neighborhood of node i.

In this paper, the subwindows in the input blurred image are considered as nodes in the CRF model similar to the formulation of discriminative random field [13]. Hence, we formulate association potential with log-likelihood of local discriminative model using the logistic function,

$$A_i(y_i, \mathbf{x}) = \log P_1(y_i|h_i(\mathbf{x}))), \tag{5}$$

where $h_i(\cdot)$ denotes the feature vector of the local region at node i and the first element is set as 1 to accommodate the bias term. The conditional probability $P_1(y_i|h_i(\mathbf{x})))$ of class y_i at node i is defined based on the logistic function:

$$P_1(y_i|h_i(\mathbf{x})) = \sigma(y_i \mathbf{w}^\top h_i(\mathbf{x})), \tag{6}$$

where $\mathbf{w}$ are parameters of the logistic function σ.

Similar to the association potential, the interaction potential is given by

$$I(y_i, y_j, \mathbf{x}) = \log P_2(y_i, y_j|\mu_{ij}(\mathbf{x})), \tag{7}$$

and

$$P_2(y_i, y_j|\mu_{ij}(\mathbf{x})) = \sigma(y_i y_j \mathbf{v}^\top \mu_{ij}(\mathbf{x})), \tag{8}$$

where $\mathbf{v}$ are the parameters of the logistic function and μ_{ij} denotes the feature vector for pair (i, j). We adopt the difference of feature vector f between node i and j, with 1 as the first element, to express μ_{ij}, $\mu_{ij}(\mathbf{x}) = [1, |h_i(\mathbf{x}) - h_j(\mathbf{x})|]^\top$. Since we do not encourage negative interaction for two nodes of different appearance or at image discontinuities, the term $\mathbf{v}^\top \mu_{ij}(\mathbf{x})$ is set to be non-negative. That is, we use the value $\max(0, \mathbf{v}^\top \mu_{ij}(\mathbf{x}))$ to substitute $\mathbf{v}^\top \mu_{ij}(\mathbf{x})$.

4.2 Image Feature

Numerous prior works have shown that smooth image regions do not provide sufficient information for kernel estimation, and instead textured regions are often selected. Nevertheless, the estimation results may still be poor even when textured regions are used [4]. Indeed, regions full of repetitive edges sometimes make no contributions to the problem when the blur movement occurs in the similar direction as the edges (see an example in the supplementary material).

To estimate blur kernels, recent algorithms focus on the use of sharp edges or edge distribution [3, 4]. Analogous to the problems with textured regions, sharp edges can be of great value for image deblurring under proper assumptions. The underlying assumption for effective use of sharp edges is that regions with high contrast in the original image maintain informative structure after motion blur. However, not all the sharp edges are effective for kernel estimation. Recently, it has been shown in [5] that edges of smaller size than the blur kernel may have adverse effect on kernel estimation, and consequently edge maps of sufficient size are used for deblurring.

Taking all these factors into consideration, we present a method to extract features from regions for image deblurring. We use the responses of a Gabor filter bank $f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})]$ to represent the oriented textures of an image region $\mathbf{x}$. Here n denotes the bin number of Gabor filters and $f_i(\mathbf{x})$ represents the proportion that the i-th orientation is the dominant direction within the observation $\mathbf{x}$. The image gradient histogram $g(\mathbf{x}) = [\text{hist}(g_1(\mathbf{x})), \text{hist}(g_2(\mathbf{x}))]$ is used to capture the distribution of edges, where $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are the first-order derivatives of the image $\mathbf{x}$ along vertical and horizontal directions.

To rule out potential negative effects from small edges, we use the mask $M(\mathbf{x}) = H(r(\mathbf{x}) - \tau)$ as [5], where $H(\cdot)$ is the Heaviside step function whose value is zero for negative argument and otherwise one, and τ is the threshold. For each pixel $p \in \mathbf{x}$, $r(p)$ measures the usefulness of gradients by

$$r(p) = \frac{\parallel \sum_{q \in \mathcal{N}_s(p)} \nabla \mathbf{x}(q) \parallel}{\sum_{q \in \mathcal{N}_s(p)} \parallel \nabla \mathbf{x}(q) \parallel + 0.5}, \tag{9}$$

where $\mathcal{N}_s(p)$ is a $s \times s$ window centered at pixel p. The feature vector $h(\mathbf{x})$ is then formed by concatenating the above-mentioned local image features,

$$h(\mathbf{x}) = [f(\mathbf{x}), g(\mathbf{x}), f(M(\mathbf{x})), g(M(\mathbf{x}))], \tag{10}$$

with varying parameters n, τ and s. We compare the proposed feature vectors with some alternatives in the supplementary material.

4.3 Parameter Learning and Inference

Let θ denote the set of parameters in the CRF model, $\theta = \{\mathbf{w}, \mathbf{v}\}$. The maximum-likelihood estimates of model parameters θ are computed with the pseudo-likelihood to approximate the partition function Z,

$$\hat{\theta} = \arg\max_{\theta} \prod_{m} \prod_{i \in S_m} P(y_i^m | \mathbf{x}^m, \mathbf{y}_{\mathcal{N}_i}^m, \theta), \tag{11}$$

where m represents the index of the training image and S_m is the graph generated from the m-th image. Based on this formulation, we have

$$P(y_i | \mathbf{x}, \mathbf{y}_{\mathcal{N}_i}, \theta) = \frac{1}{z_i} \exp(A_i(y_i, \mathbf{x}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{x})), \tag{12}$$

and the partition function can be written as

$$z_i = \sum_{y_i \in \{-1,1\}} \exp(A_i(y_i, \mathbf{x}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{x})). \tag{13}$$

To balance the effect between association and interaction potentials, we add in penalty term $\frac{1}{2\phi^2}\mathbf{v}^\top \mathbf{v}$, where the variable ϕ is pre-defined in this work. We solve the optimization problem in the log pseudo-likelihood form,

$$\begin{aligned} \hat{\theta} = \arg\max_{\theta} \sum_{m} \sum_{i \in S_m} [\log \sigma(y_i \mathbf{w}^\top h_i(\mathbf{x})) + \\ \log \sum_{j \in \mathcal{N}_i} \sigma(y_i y_j \mathbf{v}^\top \mu_{ij}(\mathbf{x})) - \log z_i] - \frac{1}{2\phi^2}\mathbf{v}^\top \mathbf{v}. \end{aligned} \tag{14}$$

We use the BFGS method to solve the optimization problem and obtain parameter $\hat{\theta}$. The loopy belief propagation (LBP) is then utilized for inference.

4.4 Good Regions to Deblur

Given a high resolution blurred image (e.g., 4000×3000 pixels), it is rather time consuming to apply deblurring algorithms with the whole image. Even with the fast deblurring algorithm [14], it may not be the best choice to use the whole image for kernel estimation for the reasons discussed in Section 4.2. We will demonstrate in Section 6 that using a region of blurred image for kernel estimation may render better deblurred result rather than using the whole image. Clearly, one immediate solution for these problems is to select a region within the input image to estimate blur kernel, and then apply a non-blind deconvolution algorithm to the whole image.

In this work, we use principal component analysis to reduce the dimensionality of the feature vectors for learning the model parameters, and LBP to classify the subwindows as good regions or not for blur kernel estimation. The label of each node (subwindow) in the training image is determined by comparing the estimated kernel and the ground truth kernel. The learning process is summarized in the supplementary material. The reason we use LBP for inference instead of graph cuts here is that we would like to obtain labels with confidence values so that it is more convenient to develop a region selection strategy. We select the top ranked subwindow to estimate kernel for simplicity although other weighted approach may be used. Given the window size, the proposed method of selecting good subwindow to deblur does not require manual selection which can be problematic and time-consuming. We show that the proposed method can effectively select the optimal subwindow for deblurring in Section 6.

Moreover, this window selection method can also be applied to non-uniform (spatially variant) blur cases which requires huge memory space and heavy computation if the whole image is used to estimate the camera motion (e.g., [14]). For instance, the non-uniform deblur algorithm [15] employs a RANSAC-based scheme to select a set of patches for estimating local blur kernels and thus rendering a good initialization. With our method, it would be easy and effective to choose a set of good patches for local kernel estimation.

5 Contribution of Feature Components

Within the model presented in Section 4, we use a feature vector consisting of several components. In this section, we determine the most important component for image deblurring via their structured output. In addition, the approach we use can also be extended to determine other effective features. We use a slightly different energy function and consider each feature component independently. That is, we decompose the associate potential $A_i(y_i, \mathbf{x})$ into a combination of feature components:

$$A_i(y_i|\mathbf{x}) = \sum_c \alpha_c \log P_1^c(y_i|h_i^c(\mathbf{x})), \tag{15}$$

where h_i^c represents different components of the feature vector, e.g., $f(x)$ and $f(M(x))$, with weights α. The conditional probability $P_1^c(y_i|h_i^c(\mathbf{x}))$ is defined based on the same logistic function as Eqn. 6. Since we aim to determine the weights of feature components rather than the parameters of the logistic functions, the parameters of logistic functions are assumed to be known (e.g., learned with a linear support vector machine). For interaction potential, we use $I(y_i, y_j, \mathbf{x}) = \beta\delta(y_i - y_j)$ to encourage labeling smoothness, where $\beta > 0$ is the weight and $\delta(\cdot)$ denotes the Kronecker function.

The objective here is to maximize the probability of labels. In the training phase, we compute the weights $\Lambda = (\alpha, \beta)$ that assign the training labels $\hat{\mathbf{y}}^m$ higher or equal probabilities than any other labels $\mathbf{y}^m$ of training image m,

$$P(\hat{\mathbf{y}}^m|\mathbf{x}^m) \geq P(\mathbf{y}^m|\mathbf{x}^m), \forall \mathbf{y}^m \neq \hat{\mathbf{y}}^m. \tag{16}$$

We can cancel the normalizer term Z from both sides of the constraints and express the constraints in terms of energies,

$$E(\hat{\mathbf{y}}^m|\mathbf{x}^m) \geq E(\mathbf{y}^m|\mathbf{x}^m), \forall \mathbf{y}^m \neq \hat{\mathbf{y}}^m. \tag{17}$$

We take a max-margin approach to compute the weights that satisfy the inequalities with the largest energy margin γ,

$$\max_{\Lambda:||\Lambda||=1} \gamma \quad \text{s. t.} \quad E(\hat{\mathbf{y}}^m|\mathbf{x}^m) - E(\mathbf{y}^m|\mathbf{x}^m) \geq \gamma, \forall \mathbf{y}^m \neq \hat{\mathbf{y}}^m. \tag{18}$$

The weights Λ are constrained to have unity norm to prevent from growing without bound. Similar to [16], we learn the weights Λ by adding slack variables and iteratively finding the minimum energy labeling with graph cuts [17]. The main steps of this algorithm are summarized in the supplementary material.

6 Experimental Results

We evaluate the inferred subwindows using three state-of-the-art deblurring algorithms [2, 9, 14] and compare the performance using the error metric introduced in [1]. This metric computes the difference between a recovered image I_r and the known ground-truth sharp image I_g, over the difference between the deblurred image I_{k_g} with the ground truth kernel k_g and the ground-truth sharp image as $||I_r - I_g||^2/||I_{k_g} - I_g||^2$. The cumulative histogram of reconstructed error ratio (briefed as cumulative error histogram for convenience) is then used to evaluate the efficiency of an algorithm. Since the dataset from [1] has limited variability of image structure and blur kernel, we use the same technique to collect 400 blurred image using 20 sharp images (with image size around 450 × 450 pixels) and 20 blur kernels for training. These kernels are generated with different orientation and shape to simulate possible blur processes. Furthermore, we construct another set of 120 challenging blurred images using 10 sharp images and 12 blur kernels for tests. Our experiments are carried out on a machine with 3.40 GHz CPU and 16 GB RAM. To infer an image of 450 × 450 pixels, it takes around 5 seconds to process with our MATLAB implementation. The source codes and data sets are available at http://faculty.ucmerced.edu/mhyang/code.

6.1 Deblurring with Good Regions

With the CRF model described in Section 4, we learn the parameters using all the 400 training images for inferring good regions to deblur. For each image, we build the graph model with overlapping subwindows of 200×200 pixels as nodes and shifts of 20 pixels. The size of subwindows is determined empirically as it is large enough to estimate the kernel of size smaller than 29×29 pixels and all the blur kernels in this data set are like that. In our experiments, there are at least 484 subwindows extracted from each image as the nodes for training. We collect the estimated kernel from each subwindow using [2] as it is one of the best kernel estimators [1]. With the estimated kernels and ground truth kernels, we label node i to be 1 or -1 using the proposed kernel similarity $S(K_i, K)$ between an estimated kernel K_i and the ground truth kernel K with the threshold λ of 0.6 (empirically set for effectiveness), where K_i represents the estimated kernel for node i. The feature parameters n, τ and s in this work are set to be 8, 0.6 to 0.9 with increment of 0.1, and 9 to 24 with increment of 4 respectively. During the inference process, the size of subwindows is set proportional to the user-defined kernel size. That is, for a larger blur kernel, the size of subwindows selected to estimate the kernel should also be increased proportionally.

Comparison with User-Selected Region. To demonstrate that user selection can be replaced by our algorithm, we compare our top inferred subwindows with user-selected regions. The users tend to choose regions with most salient edges and variances as shown in Figure 5(b)(c) and Figure 6(b)(c). The user selection strategy works well in some situations but usually requires several trials to obtain a good result. On the contrary, our proposed method does not require user guidance and the deblurred results by the inferred subwindows of our algorithm outperform that using user-selected regions.

Comparison with Region Selection Algorithms. We compare the proposed algorithm with two other region selection methods for deblurring [2, 5]. The automatic subwindow selector by [2] searches for the regions with high variance and low saturation for kernel estimation. We implement another subwindow selection method by following the idea in [5] that the image deblurring process may adversely degrade by the negative effect of small edges. In this selector, we remove small edges using the mask $M(\mathbf{x})$ and then search for the subwindow with most salient edges.

In this experiment, we select the top ranked subwindow from our inference algorithm for blur kernel estimation for simplicity although other alternatives may be used. We apply other methods to select one subwindow from each image for kernel estimation and then apply two state-of-the-art algorithms [2, 9] to recover the whole image. Figure 5 and Figure 6 illustrate the comparison using the deblurring algorithm of [2] and [9] respectively. We note that although some inferred windows (e.g., Fig. 5(e) and (f)) appear to be similar, their locations are different (60 pixels apart). The deblurring results on subwindows inferred by our algorithm are better than those from the other two subwindow selection strategies.

Comparison with Deblurring Using the Whole Image. We also compare with the deblurred results by [14] which uses the whole image for kernel estimation. In this experiment, we test two region selection strategies based on the inference results. One is to select the top ranked subwindow to estimate blur kernel, and another is to combine the top ten good subwindows. To combine the top ten subwindows, we choose the smallest rectangle which covers all the subwindows as the region for simplicity. We note that the top ranked subwindows are usually clustered due to the usage of CRF model which encourages spatial correlation, and thus the rectangular region is still of small size compared with the whole image as shown in Figure 7. Compared to the results obtained from the whole images in Figure 7, our algorithm with inferred region generates comparable or superior kernel estimation and reconstructed images.

Quantitative Comparison. We conduct extensive comparisons using 120 challenging test images and present their cumulative error histograms. Given an inferred region by the proposed algorithm, we use the fast algorithm [14] to estimate a blur kernel, and the non-blind deconvolution algorithm [9] to recover the latent image (similar as [14]). For thorough evaluations, we compare the results using the above-mentioned region selection strategies [2] and the whole image [14]. In addition, we compare with [5] using a region or the whole image for kernel estimation. As shown in Figure 4, the curve using our top ten subwindows generally performs better than other algorithms [5, 14] using the whole image. The results also show that not all the information in a blurred image is useful and using the whole image for kernel estimation may not be the best choice. We note that the reconstructed results are visually plausible even when the error ratio is around 5 (an example is provided in the supplementary material), which is different from the observations in [1]. The reason is that we employ different non-blind deconvolution algorithms and larger test images.

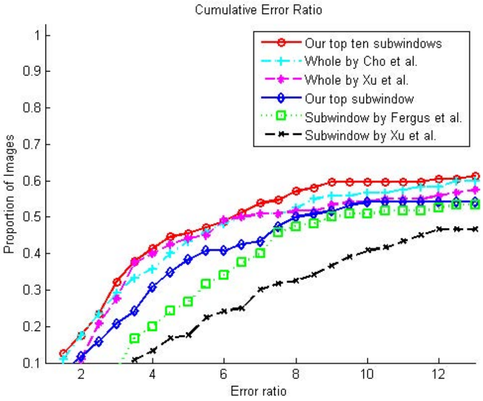
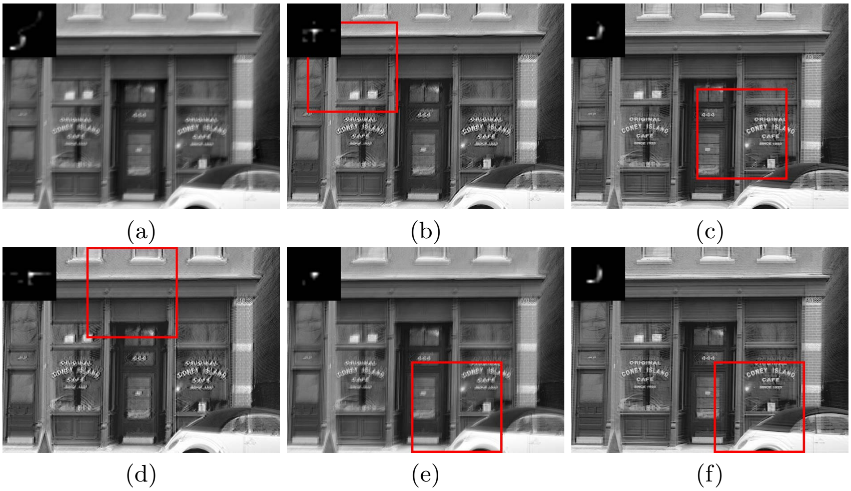
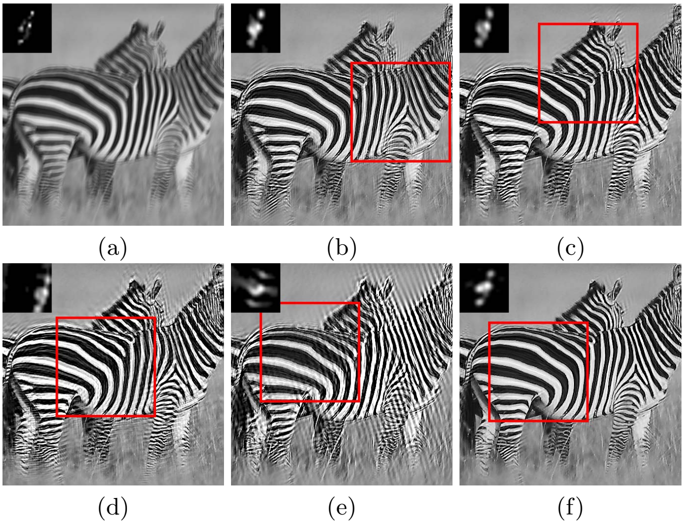


Fig. 4. Cumulative error histograms on deblurred results with different window selection methods including ours, [2] and [5]. We use the algorithm in [14] for kernel estimation and the algorithm in [9] for non-blind deconvolution.


(a) (b) (c)

(d) (e) (f)

Fig. 5. Comparison on different window selection strategies for deblurring using [2]. (a) input blurry image and corresponding kernel; (b) & (c) user-selected subwindows (red box) and the deblurred results; (d) subwindow selected by window selector [2] and the deblurred result; (e) subwindow selected by window selector following [5] and the deblurred result; (f) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display).


(a) (b) (c)

(d) (e) (f)

Fig. 6. Comparison on different window selection strategies for deblurring using [9]. (a) input blurry image and corresponding kernel; (b) & (c) user-selected subwindows (red box) and the deblurred results; (d) subwindow selected by window selector [2] and the deblurred result; (e) subwindow selected by window selector following [5] and the deblurred result; (f) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display).

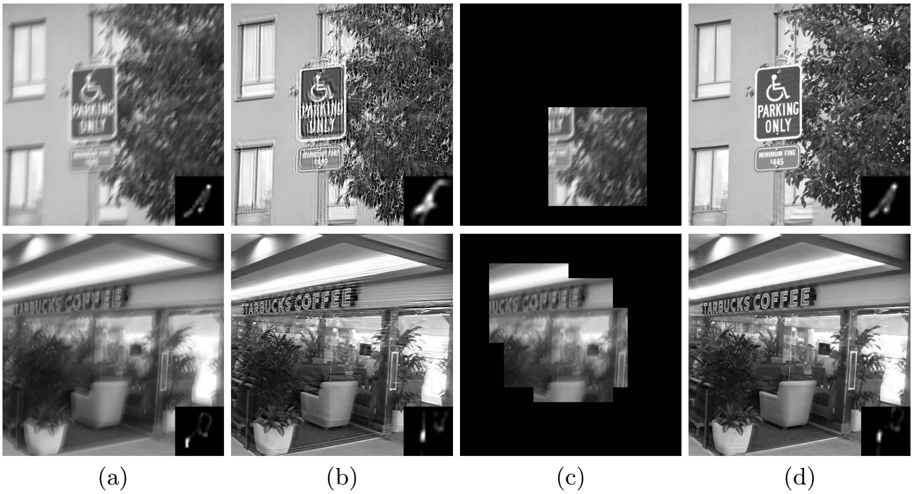


Fig. 7. Inferred regions for deblurring using [14]. The first row shows results with top ranked subwindow and the second row with top ten subwindows. (a) input image and the ground truth blur kernel; (b) kernel estimation and deblurred result using the whole image [14]; (c) inferred good regions by our algorithm; (d) deblurred results using the inferred good regions (best viewed on high-resolution display).

Analysis. Our empirical results also provide some insights about the kinds of image structures that are favored by different deblurring algorithms. In the *zebra* image, the selected window by our algorithm contains relatively strong edges, as suggested in [1, 2]. However, small or detailed edges may not help kernel estimation as shown in Figure 6(b). Furthermore, the subwindow with various oriented edges is selected rather than the regions with repetitive textures. The empirical results and observations bear out our intuition and analysis in Section 4.2.

6.2 Feature Components

With the model described in Section 5, we learn the weights Λ from a set of 400 images. The results show that the weights α_c for feature components $f(\mathbf{x})$ and $f(M(\mathbf{x}))$ of Eqn. 10 have the highest values. This indicates that Gabor filter responses are more discriminative for this task, while using the original blur image is more effective than using the image after removing small edges.

7 Conclusions

Recent works focus on introducing priors or additional information to facilitate deblurring, and considerably less efforts are made to study structural information of the input blurred image itself. We address this issue and exploit informative image structure for deblurring. In this paper, we propose a learning-based approach for selecting good features and good patches for blur kernel estimation.

Towards this, we introduce a kernel similarity metric for effectively comparison between kernels. The proposed algorithms select good regions automatically for deblurring with efficient and favorable results, thereby relieving users from tedious trials for selecting image patches.

References

1. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR, pp. 1964–1971 (2009)
2. Fergus, R., Singh, B., Hertzmann, A., Rowels, S.T., Freeman, W.T.: Removing camera shake from a single photograph. In: ACM SIGGRAPH, pp. 787–794 (2006)
3. Joshi, N., Szeliski, R., Kriegman, D.J.: PSF estimation using sharp edge prediction. In: CVPR (2008)
4. Cho, T.S., Joshi, N., Zitnick, C.L., Kang, S.B., Szeliski, R., Freeman, W.T.: A content-aware image prior. In: CVPR, pp. 169–176 (2010)
5. Xu, L., Jia, J.: Two-Phase Kernel Estimation for Robust Motion Deblurring. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 157–170. Springer, Heidelberg (2010)
6. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML, pp. 282–289 (2001)
7. Levin, A.: Blind motion deblurring using image statistics. In: NIPS, pp. 841–848 (2006)
8. Jia, J.: Single image motion deblurring using transparency. In: CVPR (2007)
9. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. In: ACM SIGGRAPH, pp. 73:1–73:10 (2008)
10. Cai, J., Ji, H., Liu, C., Shen, Z.: Blind motion deblurring from a single image using sparse approximation. In: CVPR, pp. 104–111 (2009)
11. Ben-Ezra, M., Nayar, S.: Motion deblurring using hybrid imaging. In: CVPR, pp. 657–664 (2003)
12. Yuan, L., Sun, J., Quan, L., Shum, H.: Image deblurring with blurred/noisy image pairs. In: ACM SIGGRAPH (2007)
13. Kumar, S., Hebert, M.: Discriminative random field. IJCV 68, 179–201 (2006)
14. Cho, S., Lee, S.: Fast motion deblurring. In: SIGGRAPH Asia (2009)
15. Gupta, A., Joshi, N., Zitnick, L., Cohen, M., Curless, B.: Single Image Deblurring Using Motion Density Functions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 171–184. Springer, Heidelberg (2010)
16. Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs Using Graph Cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)
17. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23, 1222–1239 (2001)