

# Kernel Fusion for Better Image Deblurring

Long Mai  
Portland State University  
mtlong@cs.pdx.edu

Feng Liu  
Portland State University  
fliu@cs.pdx.edu

## Abstract

*Kernel estimation for image deblurring is a challenging task and a large number of algorithms have been developed. Our hypothesis is that while individual kernels estimated using different methods alone are sometimes inadequate, they often complement each other. This paper addresses the problem of fusing multiple kernels estimated using different methods into a more accurate one that can better support image deblurring than each individual kernel. In this paper, we develop a data-driven approach to kernel fusion that learns how each kernel contributes to the final kernel and how they interact with each other. We discuss various kernel fusion models and find that kernel fusion using Gaussian Conditional Random Fields performs best. This Gaussian Conditional Random Fields-based kernel fusion method not only models how individual kernels are fused at each kernel element but also the interaction of kernel fusion among multiple kernel elements. Our experiments show that our method can significantly improve image deblurring by combining kernels from multiple methods into a better one.*

## 1. Introduction

Images taken by consumer photographers often come out blurry due to camera shake caused by the sensor movement during the exposure time. In many cases, the problem is inevitable without using equipments like tripods, which are often not available. Removing the effect of camera shake from images has long been an important research topic in computer vision and graphics.

Under the assumption of white noise and spatially invariant blur, the image blur process can be modeled mathematically as image convolution

$$B = k * I + n \quad (1)$$

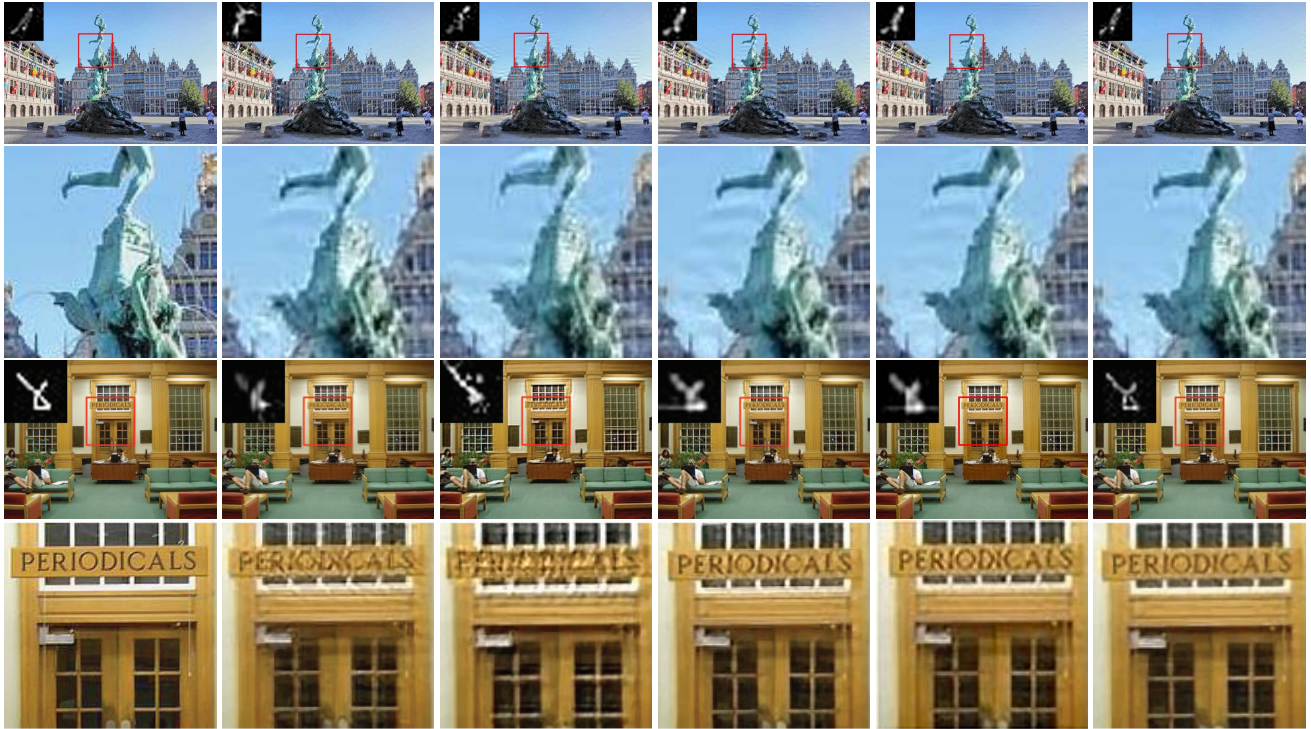
where  $*$  is the convolution operator.  $B$  is the acquired blurry image,  $I$  is the sharp image,  $k$  is the convolution kernel representing the blurring process, and  $n$  is the noise. The goal of image deconvolution is to recover the sharp image  $I$  given its blurry observation  $B$ .

There is now a rich literature on image deconvolution [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 35, 36, 37, 38, 39]. In this paper, we focus on *blind image deconvolution* where the blur kernel  $k$  is unknown and needs to be recovered along with the sharp image. Blind deconvolution is a severely under-constrained problem. Given an observed blurry image  $B$ , there are infinitely many pairs of  $k$  and  $I$  that satisfy Equation 1.

It has been noted by Levin *et al.* that estimating accurate blur kernel is a critical component in image deblurring [19]. Over the past decade, remarkable research effort has been devoted to developing methods that can accurately recover blur kernels [4, 5, 7, 8, 14, 18, 23, 28, 30, 32, 34]. Levin *et al.* provide an iterative framework to approximate the maximum posterior inference for the blur kernel  $k$  given the blurry input  $B$ , marginalizing over all possible latent images  $I$  [18]. Recent research directly exploits information from the blurry image to estimate blur kernels. Joshi *et al.* [14] and Cho *et al.* [4] predict the sharp edges from the blurry image and use the predicted edge profile to constrain kernel estimation. Sun *et al.* employs patch-based priors to focus on the image regions that are more informative for blur kernel estimation [30]. Cho *et al.* construct the Radon transform of the blur kernel from the blurry image and use that to obtain the kernel estimation [5].

While existing blind deblurring methods have shown significant improvement in terms of kernel estimation accuracy (and thus the deblurring results), their performance can still be improved. As illustrated in Figure 1, while each method can estimate good kernels for some images, none of the methods can perfectly recover the kernels in all the cases. More interestingly, we observed that as different methods often incorporate different prior knowledge into their deblurring frameworks, they often complement each other. We believe that combining multiple kernels from different methods can lead to a better kernel, as illustrated in Figure 1 and Figure 6.

This paper presents a method to leverage existing blur kernel estimation methods to better support image deblurring than these existing methods themselves. Our idea is to



(a) Ground-Truth (b) Cho et al.[4] (c) Goldstein et al.[8] (d) Xu et al.[32] (e) Shan et al.[28] (f) Kernel Fusion

Figure 1: Kernel fusion and image deblurring examples. By combining multiple kernel estimations from different image deblurring methods, our kernel fusion method can produce the final kernel that is more accurate and leads to better deblurring results than each individual one.

fuse kernels from multiple existing deblurring methods such that the combined kernel outperforms each individual one. The problem is challenging in that the fusion process needs to capture the complex relation among individual estimations, as well as how they relate to the underlying true blur kernel. Classical fusion methods such as (weighted) averaging cannot lead to good fusion results, as shown in Section 2.1. Therefore, we develop data-driven approaches to kernel fusion that learns how individual kernels contribute to the final fusion result and how they interact with each other. After examining various kernel fusion models, we find that kernel fusion using Gaussian Conditional Random Fields (GCRF) performs best. This GCRF-based kernel fusion method not only models how individual kernels are fused at each kernel element but also the interaction of kernel fusions among multiple elements.

This paper has the following contributions. First, we introduce the novel idea of kernel fusion. Our goal is to combine multiple kernel estimations such that the fusion result can outperform each individual one. Therein, we develop data-driven approaches which can effectively learn the good kernel fusion models from training data. We demonstrate that our method can effectively combine different kernel estimations into a better one. Finally, we collect a large image deblurring benchmark, which will be made publicly available to the research community.

## 2. Kernel Fusion

Given  $N$  kernel estimations  $\{k_i\}_{i=1..N}$  generated by  $N$  different kernel estimation methods, the goal is to combine them into the final kernel  $k^*$  that is more accurate than each individual kernel and better supports image deblurring.

$$k^* = f_c(\{k_i\}_{i=1..N}) \quad (2)$$

where the function  $f_c$  denotes the combination function that performs the kernel fusion. In this section, we examine various strategies to construct the fusion model  $f_c$ . We then develop our data-driven approach to effectively learn the combination model  $f_c$  from training data.

### 2.1. Baseline Method: Kernel Averaging

To the best of our knowledge, we are the first to attempt the problem of combining multiple kernel estimations. To serve as a baseline method, and to motivate the challenge of our problem, we consider the classical and widely used fusion strategy: estimation averaging. Specifically, this combination strategy takes the average of all individual kernel estimations to obtain the fusion result.

$$k_{AVG}^* = \frac{1}{N} \sum_{i=1}^N k_i \quad (3)$$

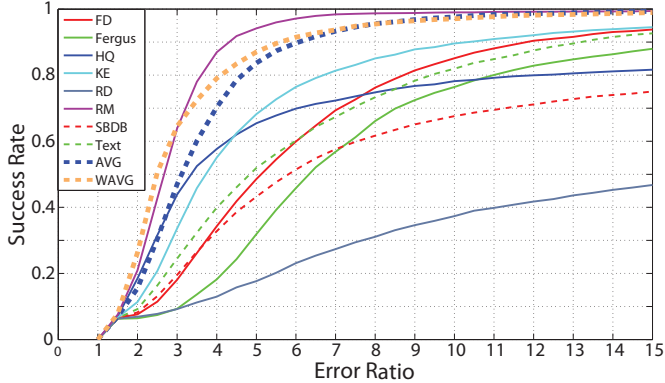


Figure 2: Classical fusion method performance. Classical fusion methods such as kernel averaging (AVG) and weighted kernel averaging (WAVG) apply the fixed combination function regardless of the input. When applying in kernel fusion, those fusion models perform worse than the best individual method in the dataset (RM).

Considering each individual kernel as a noisy estimation of the true kernel, estimation theory suggests that averaging all individual kernel has the advantage of reducing the noise in individual kernels. However, this method has an important limitation. Since simple averaging considers each individual kernel equally, the bad estimation result from some method may compromise the good one.

We also experimented with the weighted average fusion strategy. The weight of each individual kernel in the fusion is determined by its quality over the whole image set.

$$k_{WAVG}^* = \sum_{i=1}^N w_i k_i \quad (4)$$

where  $w_i$  is determined by the percentage of images in the image set where the  $i^{th}$  deblurring method works the best among  $N$  individual methods.

To evaluate the performance of the two classical kernel fusion strategies AVG and WAVG described above, we compared them against each of the eight individual kernel estimation methods by comparing their deconvolution results. Specifically, we collect a large set of synthetically blurred images (for which the ground-truth sharp images and kernels are available) and use the non-blind deconvolution algorithm in [17] along with each kernel estimation method to generate its deblurring result for each image. We evaluate the deconvolution results from each method according to its error ratio (ER) curve ([19]) which measures the percentage of images that the kernel estimation can generate successful deconvolution results at a range of error levels (Section 3.2). In general, the higher the ER curve, the better the overall deconvolution performance.

Figure 2 shows the ER curves for the deconvolution results obtained from eight different kernel estimation meth-

ods as well as those from the kernel averaging fusion (AVG) and weighted kernel averaging fusion (WAVG) strategies. The curves show that the AVG kernel fusion strategy leads to worse results than the best individual method (the RM method in this case). In addition, while the WAVG fusion strategy can improve the kernel fusion quality over AVG by weighing each individual method in the fusion, it is still not able to outperform the best individual method.

These results show the limitation of the classical fusion models in combining kernel estimation results. These models apply a pre-defined combination function for every input. However, in our context, the fusion model should adapt to different input, because the relation among individual kernel estimations as well as the relation between the estimated kernel and true underlying kernel is likely to change from image to image. We argue that a good kernel fusion model should be flexible enough to model these complex relations. In this paper, we develop data-driven approaches to learn such a fusion model from training data.

## 2.2. Element-Wise Kernel Fusion

In this method, we predict the value at each kernel element independently. Specifically, we associate each kernel element  $k(p)$  with a vector composed of the corresponding kernel element from each individual kernel estimation  $f_p = \{k_i(p)\}_{i=1..N}$ .

The problem of learning the combination function  $f_c$  becomes a classical regression problem: learn the mapping function that maps the feature vector  $f_p$  to the corresponding true kernel value at element  $p$ .

In this method, we model the mapping function  $f_c$  using the Random Forest regression technique [2] (RF), which is one of the state-of-the-art regression models. We train the RF model to map individual kernel values at each element to its true kernel value. The RF model is trained using the training data where the ground-truth kernel is available.

After training, the trained RF model can be used to perform element-wise kernel fusion for any new given image. Specifically, given a new blurry image and the set of  $N$  individual kernel estimations generated from it using different deblurring methods, we form the feature vector  $f_p$  at each kernel element  $p$  as described above. We then use the trained RF model to predict the kernel value at each kernel element from its feature vector through the RF prediction procedure.

In this paper, we use the Random Forest implementation for MATLAB from Jaiantilal et al. <sup>1</sup>

## 2.3. Kernel Fusion using Gaussian Conditional Random Fields

While the above element-wise fusion strategy can generate promising kernel fusion results (as demonstrated in Sec-

<sup>1</sup><https://code.google.com/p/randomforest-matlab/>

tion 3.2), it has an inherent limitation. As the element-wise kernel fusion model is trained to predict the kernel value at each kernel element individually, it does not capture the relationship between the values of adjacent kernel elements. Physically, the blur kernel represents the projection of the camera path during the exposure time. That implies strong spatial relationship among neighboring kernel elements.

Our second method addresses this problem by modeling kernel fusion using a Gaussian Conditional Random Fields (GCRF) framework. Specifically, we model kernel elements as nodes in a graph. Like the element-wise fusion model, we associate each node  $p$  with an individual-kernel-value features  $f_p = (k_1(p), k_2(p), \dots, k_N(p))$ .

We use the GCRF with an eight-neighboring system to model the relationship between the kernel fusion value between neighboring kernel elements. In our GCRF framework, the conditional distribution of the fused kernel given all individual kernels is modeled as

$$P(k|\{k_i\}_{i=1..N}) \propto \exp(-E(k|\{k_i\}_{i=1..N})) \quad (5)$$

where the energy function is composed of the collection of local energy terms defined at each node and each pair of neighboring nodes

$$E(k|\{k_i\}_{i=1..N}) = \sum_{p \in k} E_u(k(p)|\{k_i(p)\}_{i=1..N}) + \sum_{p \in k, q \in N(p)} E_s(k(p), k(q)|\{k_i(p), k_i(q)\}_{i=1..N}) \quad (6)$$

where  $N(p)$  denotes the set of neighboring nodes to  $p$ .

The unary term  $E_u$  in Equation 6 models how the kernel value at each element in the final kernel estimation depends on the kernel value from each individual kernel. We model the unary term as a local Gaussian distribution

$$E_u(k(p)|\{k_i\}_{i=1..N}) = \frac{1}{2} \theta_p (k(p) - \mu_p^T f_p)^2 \quad (7)$$

where  $\theta_p$  and  $\mu_p^T f_p$  represents the precision (i.e. inverse of variance) and the mean of the local Gaussian model.

In addition, our GCRF model also model the interaction between neighboring kernel elements through the pair-wise local energy term

$$E_s(k_{pq}|\{k_i\}_{i=1..N}) = \frac{1}{2} (k_{pq} - \Phi_{pq} f_{pq})^T \Theta_{pq} (k_{pq} - \Phi_{pq} f_{pq}) \quad (8)$$

where  $k_{pq} = [k(p), k(q)]$  is the fused kernel values at elements  $p$  and  $q$ .  $f_{pq}$  is the concatenation of the node feature

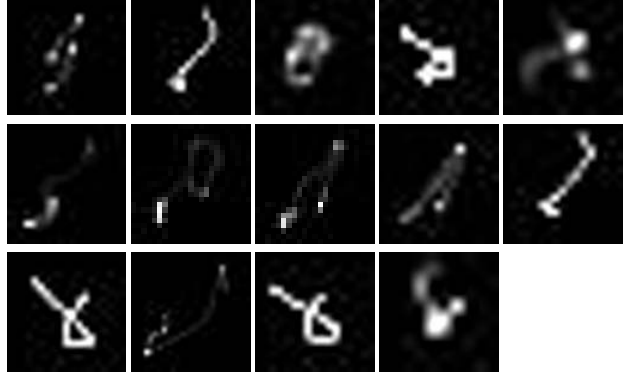


Figure 3: The set of 14 kernels used in generating our synthetically blurred dataset.

vectors  $f_p$  and  $f_q$ . Similar to the unary term, the estimated kernel values at the pair of neighboring elements  $k_{pq}$  is governed by a local Gaussian model with mean  $\Phi_{pq} f_{pq}$  and precision matrix  $\Theta_{pq}$ .

We follow the Regression Tree Field (RTF) framework [25] (which provides a powerful GCRF model instantiation) to make the model parameters  $\{\theta_p, \mu_p, \Theta_{pq}, \Phi_{pq}\}_{p \in k, q \in N(p)}$  depend on the input data  $\{k_i\}_{i=1..N}$  via the local feature vectors  $f_p$  and  $f_{pq}$  at each local energy term.

RTF framework has recently been applied successfully in various challenging computer vision problems, such as image denoising [13] and non-blind image deblurring [26]. In this paper, we apply the RTF model to our problem of kernel fusion. We train the model parameters by minimizing the difference between the kernel fusion results and the ground-truth kernels over the training data. Once the model parameters have been trained, we can use them in the standard GCRF inference procedure to compute the kernel fusion  $k^*$  for any new image as

$$k^* = \operatorname{argmin} E(k|\{k_i\}_{i=1..N}) \quad (9)$$

where the energy function  $E$  is defined as in Equation 6.  $\{k_i\}_{i=1..N}$  denotes the set of kernel estimation generated from the input image using each individual deblurring methods as before. In our implementation, we perform both the RTF training and testing using the available RTF-1.1 software package provided by [26]<sup>2</sup>.

## 3. Experiments

### 3.1. Dataset

As our method relies on a large amount of training data to train the kernel fusion models, we established a large synthetically blurred image set. To generate the dataset, we

<sup>2</sup><http://www.gris.tu-darmstadt.de/research/visinf/software/index.en.htm>

first collect 289 sharp images from online sources, including Flickr, Facebook, and Google Plus. These images span a wide variety of scene content (e.g. indoor, outdoor, natural scene, man-made environment, people, animal). We synthetically blur each image in the dataset using 14 different blur kernels. In particular, we make use of eight kernels provided by [19] and six kernels extracted from the benchmark [16]. As the benchmark in [16] consists of spatially varying blur, we extracted the kernels by randomly select different local regions in their ground-truth kernels. Figure 3 shows all 14 kernels used in generating our dataset. Like [30], we add additive Gaussian noise to each blurred image to simulate the sensor noise.

To serve as individual methods for our kernel fusion framework, we make use of eight state-of-the-art deblurring methods : FD (Cho et al.[4]), FG (Fergus et al.[7]), HQ (Shan et al.[28]), KE ( Goldstein et al.[8]), RD (Cho et al.[5]), RM (Xu et al.[32]), SBDB (Zhang et al.[35]), and TX (Pan et al.[23]). For each blurred image in our dataset, we use the author provided implementation of each method to generate a blur kernel estimation, resulting in eight individual kernel estimations for each image. In our experiment, we assume the size of the kernel is known and use the ground-truth kernel size as the input to each deblurring method. All other parameters (if there is any) for each of the deblurring methods are kept as the default setting provided by the authors’ implementations.

We note that occasionally some deblurring methods fail to produce deblurring results for some blurring images due to program crashing. We only keep the images on which all eight methods successfully produce kernel estimation results. In total, we keep 2,976 blurring images in our dataset. Each blurring image has 8 kernels estimated using the above 8 different methods.

For each of the experiments in this section, we randomly select 1,000 images for training and 1,976 images for testing. For each experiment, we repeat the random partition ten times and report the average results. Please refer to our project website <sup>3</sup> for more results.

### 3.2. Kernel Fusion Performance

We evaluate the effectiveness of our fusion method by comparing the quality of kernel estimation from our method against that from each individual methods. Specifically, we evaluate the quality of a kernel estimation result according to both its similarity to the true kernel as well as the visual quality of its resulted deconvolution result.

#### 3.2.1 Similarity to Ground-Truth Kernels

We first compare different methods according to their ability to accurately recover the true underlying kernel. To eval-

<sup>3</sup><http://graphics.cs.pdx.edu/project/kernelfusion>

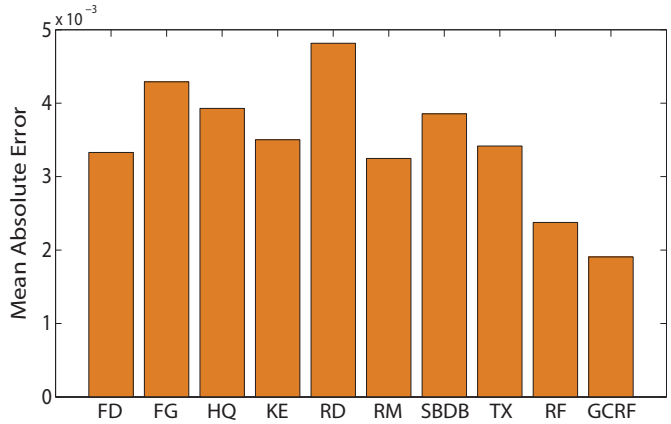


Figure 4: Mean Absolute Error (MAE) to ground-truth kernels. Our data-driven kernel fusion methods (RF and GCRF) can produce kernel estimation results which can better resemble the true kernel compared to each individual method.

uate how well a kernel estimation method can recover the true underlying blur kernel, we measure how similar the estimated kernel is to the ground-truth kernel. Specifically, we measure the similarity to the ground-truth kernel using the mean absolute error (MAE) between the two kernel maps.

$$MAE(k|k_g) = \frac{1}{|k|} \sum_{p \in k} |k(p) - k_g(p)| \quad (10)$$

where  $k$  denotes the estimated kernel,  $k_g$  denotes the ground-truth kernel, and  $|k|$  denotes the number of elements in the kernel  $k$ .

We compare the kernel estimation quality from different methods according to their average MAE over all images in the testing dataset. Figure 4 shows the average MAE error over the testing data for each kernel estimation method. We can observe from the figure that by learning the fusion model directly from the training data, the kernel estimations from our kernel fusion methods can better resemble the ground-truth kernels significantly than each individual method.

#### 3.2.2 Deconvolution Result Quality

The main goal of image deblurring is to recover the sharp image. Therefore, the quality of an estimated blur kernel should also be evaluated according to the visual quality of the deconvolution result generated by a non-blind deconvolution method with that kernel. Following [30], we make use of different metrics in assessing the deconvolution results : the *Peak-Signal-to-Noise Ratio (PSNR)*, the *Structure Similarity (SSIM)*, and the *Error-Ratio (ER) curve*. To make a fair comparison among kernel estimation results from different deblurring methods, we generate the deconvolution

Method	PSNR	SSIM
FD (Cho et al.[4])	22.33	0.681
FG (Fergus et al.[7])	20.76	0.608
HQ (Shan et al.[28])	20.39	0.670
KE (Goldstein et al.[8])	23.36	0.717
RD (Cho et al.[5])	15.45	0.515
RM (Xu et al.[32])	25.34	0.794
SBDB (Zhang et al.[35])	19.76	0.646
TX (Pan et al.[23])	22.46	0.692
Ours (RF)	25.87	0.804
Ours (GCRF)	<b>26.75</b>	<b>0.829</b>

Table 1: Visual quality measurement of deblurring results generated from each estimated kernel. Compared to each individual deblurring method, our kernel fusion methods can more accurately recover the underlying kernel and thus can significantly improve the final deconvolution results.

results from each kernel estimation using the same non-blind deconvolution algorithm [17].

**PSNR and SSIM** are two quality metrics that are widely used in image processing and computer vision research to measure the image reconstruction quality. These two metrics directly measure the visual difference between the reconstruction results (i.e. the deconvolution results in this paper) and the corresponding ground-truth images.

Table 1 shows the average PSNR and SSIM quality measurement for all methods over the testing data. The result indicates that by combining multiple kernel estimation results, our methods can predict the final kernel estimation that results in better deconvolution results than each individual method.

**Error Ratio curve:** In [19], Levin *et al.* noted that a wider kernel tends to result in larger deconvolution error even with the true kernel. They suggested normalizing this effect by measuring the ratio between the deconvolution error with the estimated kernel and that with the true kernel using the same non-blind deconvolution method.

$$ER(k|k_g) = \frac{|I_k - I_g|_2}{|I_{k_g} - I_g|_2} \quad (11)$$

where  $I_g$  denotes the ground-truth sharp image.  $I_k$  and  $I_{k_g}$  denotes the deconvolution results with the estimated kernel  $k$  and the deconvolution result with the ground-truth kernel  $k_g$  respectively. For evaluation, we compute the ER curve (Success Rate vs. Error Ratio) for each method. At each error ratio level, the success rate of the method represents the percentage of test images that has the error ratio below that error level.

Figure 5 shows the ER curves generated from different methods. This result is consistent with that in Table 1. Our data-driven kernel fusion method can effectively combine multiple kernel estimation into the final kernel estimation that outperforms each individual one. Among our two

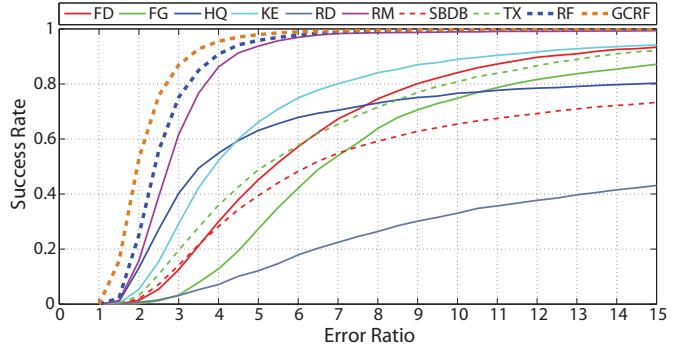


Figure 5: Error ratio curves. Our kernel fusion methods lead to kernel estimation results with significantly higher success rates than each individual method.

kernel fusion approaches, our GCRF-based fusion model can better capture the spatial relationship between neighboring kernel element values and therefore consistently improve the performance over our element-wise (RF) model by a large margin. In the remaining experiments, we will only experiment with our GCRF-based fusion method.

Figure 6 shows some qualitative results comparing the deblurring quality of the individual methods and the kernel fusion results. We can see from these examples that our fusion model can effectively combine multiple kernel estimations such that the combined result represent the true kernel more accurately compared to each individual one. That in turn leads to better final deblurring results. Overall, our method can often produce sharper deconvolution results with less ringing artifacts.

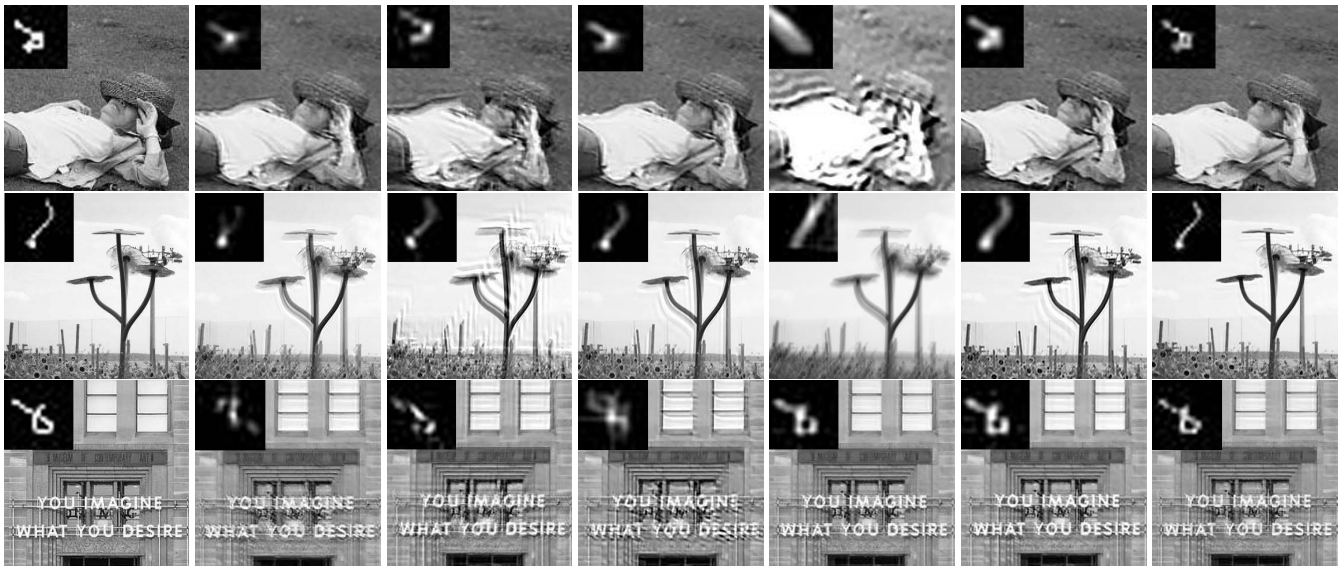
### 3.2.3 Generalization Performance

Previous experiments demonstrate the effectiveness of our data-driven fusion methods. Leveraging the training data, our method is able to learn the relation between the true underlying kernels and their noisy estimation from individual deblurring method and effectively capture that relation in the resulted fusion model.

We further evaluate the robustness of our method in the case where the true underlying kernel is different from those used for training. we perform an additional experiment. In this experiment, for each blurred image in the dataset, we train our fusion models using training images blurred by the different kernels, and use the trained fusion model to produce the final kernel estimation on that image.

Figure 7 shows the comparison among different methods according to the ER curves. We note that this curve is consistent with that in Figure 5, which demonstrates the robustness of our kernel fusion method in generalizing to new blur kernels.

To further evaluate how our kernel fusion method generalize for images that are different on those that are used to train the fusion model, we perform the experiment on



(a) Ground-Truth (b) Cho et al.[4] (c) Goldstein et al.[8] (d) Pan et al.[23] (e) Shan et al.[28] (f) Xu et al.[32] (g) Kernel Fusion

Figure 6: Qualitative comparison of deblurring results using different kernel estimation. By combining multiple kernel estimations from different image deblurring methods, our kernel fusion can produce the final kernel estimation that better resemble the ground-truth kernel and leads to better deblurring results compared to each individual one.

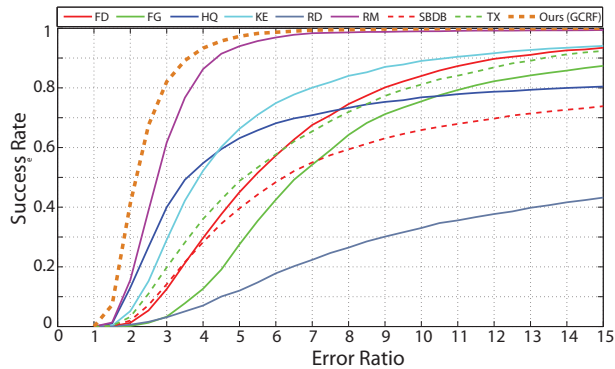


Figure 7: Generalization to kernels that does not appear during training. The ER curves show that our kernel fusion method can outperform each individual method even when the testing blurry image was blurred with completely different kernels than the ones our model was trained on.

the existing deblurring dataset provided by [30], which has no overlap with our dataset. We train the fusion models on our dataset and test on the new dataset (We only consider the images where all eight individual methods can generate kernel estimation results without crashing, which results in 450 images). From Fig. 8 shows the result consistent with that from the previous experiment: kernel fusion can improve the deblurring results over each individual one.

**Performance on real blurred images :** We test our method on a number of real blurred images collected from previous papers [16, 28, 7, 9] as well as those collected by ourselves. Visually examine the results, we observe that when some individual methods can generate reasonable deblurring results, kernel fusion can take advantage of those

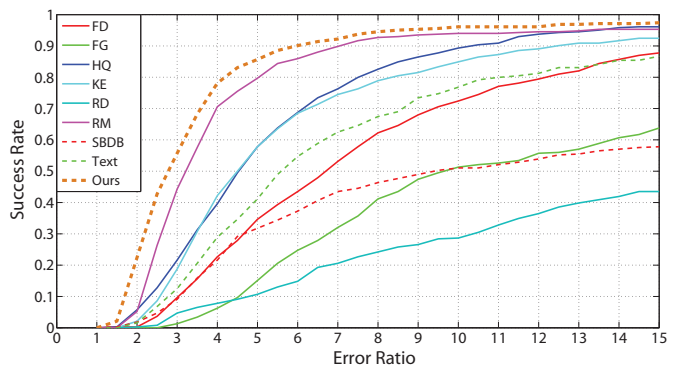


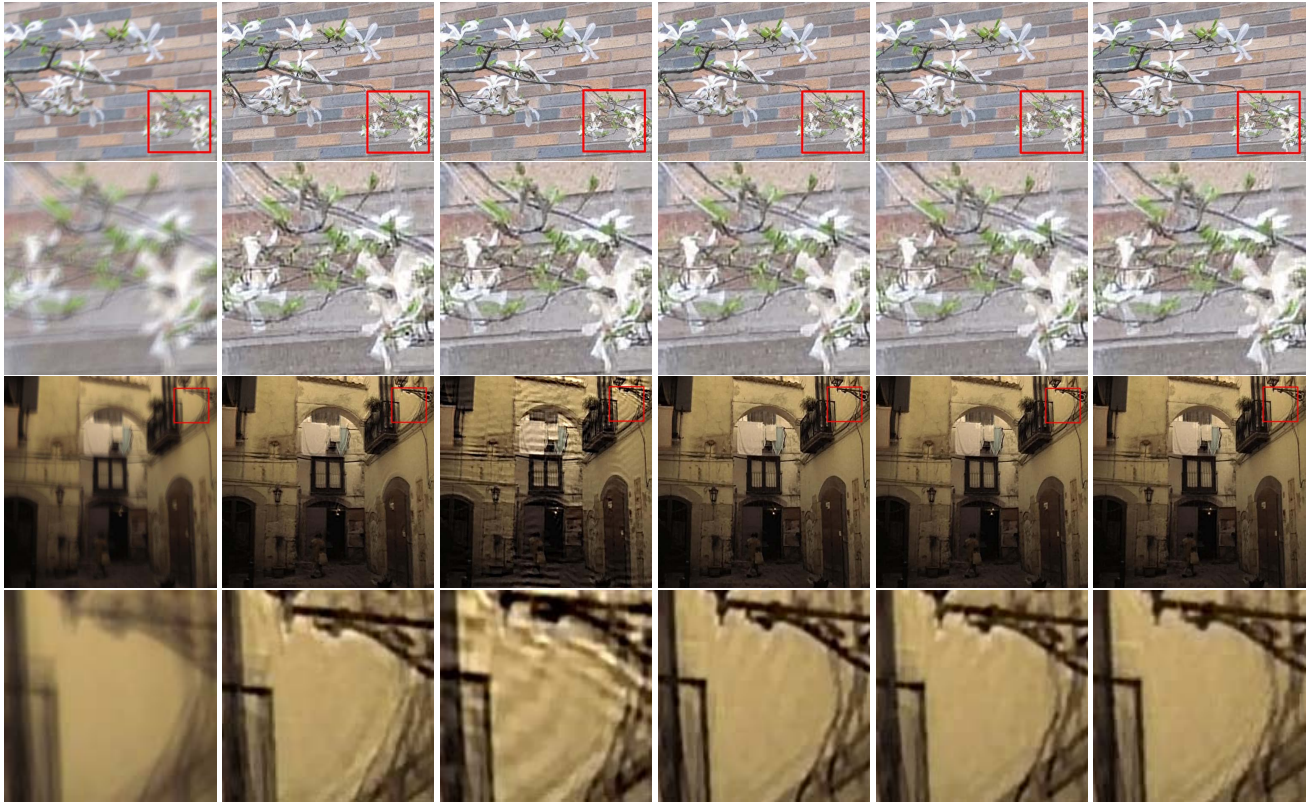
Figure 8: Generalization to images from the dataset provided by [30], which has no overlap with our dataset. Our kernel fusion method can outperform each individual method on the new images.

good methods to generate a better deblurred image. We show two examples in Fig. 9.

### 3.3. Discussion

Our work provides a framework to combine multiple blur kernel estimations into a better one. An alternative approach in leveraging a set of available kernel estimation results is to select the best one among those to use as the overall result. It is interesting to examine how effective that approach is compared to our kernel fusion approach.

To make the comparison, we consider the *Oracle Best-Kernel-Selection* method. Given a blurry input image and a set of  $N$  individual kernel estimation results, we use the ground-truth image to measure the quality of the deconvolution results and pick the best individual kernel. As this *Ora-*



(a) Input (b) Cho et al.[4] (c) Goldstein et al.[8] (d) Xu et al.[32] (e) Shan et al.[28] (f) Kernel Fusion

Figure 9: Qualitative examples on real blurred images. When a few individual methods can generate reasonable deblurring results, kernel fusion can take advantage of those good methods to generate better results.

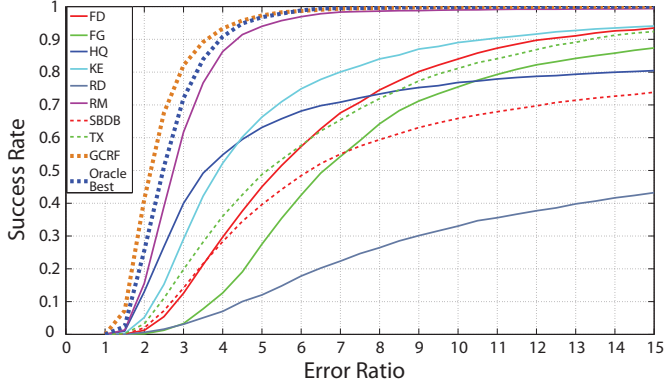


Figure 10: Comparison between kernel fusion and best-kernel-selection.

the *Best-Kernel-Selection* method uses the ground-truth image in selecting the best individual kernel estimation result for each input, its performance is in fact the upper bound of that from any possible best-kernel-selection method.

Figure 10 compares the deconvolution performance of our kernel fusion method to that of the oracle best-kernel-selection approach. This result shows the advantages of our kernel fusion approach. While selecting the input-dependent best kernel from the set of individual kernel estimations can improve over each individual one, combining those kernels can even further improve the performance.

**Limitation:** Our experiments show that our kernel fusion method can consistently improve the kernel estimation performance of each individual image deblurring method. However, we note that there is a limit of improvement. Because our kernel fusion framework depends solely on the kernel estimation results of individual method, when all the method severely fail to estimate the true kernel, our method may fail too. That may happen especially where the blurring process of the input image significantly violates the uniform blur assumption made by all individual methods used by our fusion model. In future work, we plan to extend our method to handle non-uniform blur.

## 4. Conclusion

In this paper, we introduce the novel approach of kernel fusion for image deblurring. Specifically, we develop data-driven approaches to kernel fusion and find that the Gaussian Conditional Random Field based fusion model works the best. By effectively combining multiple kernel estimation together, our kernel fusion method can consistently outperform each individual deblurring method in estimating blur kernels, and lead to superior deconvolution results. As image deblurring research has been progressing rapidly, our work provides a principled framework to leverage the advantages of existing methods to push forward the state-of-the-art in image deblurring.



**Acknowledgement.** The source images in Figure 1 and Figure 6 are used from Flickr users, including vanherdehaage, Francisco Antunes, Mr Hicks46, and See-ming Lee, under a Creative Commons license. This work was supported in part by NSF grants IIS-1321119, CNS-1205746, and CNS-1218589.

## References

- [1] M. Ben-Ezra and S. K. Nayar. Motion Deblurring Using Hybrid Imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–664, 2003. 1
- [2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001. 3
- [3] J. Chen, L. Yuan, C.-K. Tang, and L. Quan. Robust dual motion deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1
- [4] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28(5):145:1–145:8, Dec. 2009. 1, 2, 5, 6, 7, 8
- [5] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman. Blur Kernel Estimation Using the Radon Transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1, 5, 6
- [6] F. Couzinie-Devy, J. Sun, K. Alahari, and J. Ponce. Learning to Estimate and Remove Non-uniform Image Blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1
- [7] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, July 2006. 1, 5, 6, 7
- [8] A. Goldstein and R. Fattal. Blur-Kernel estimation from spectral irregularities. In *Proceedings of the European conference on Computer Vision*, 2012. 1, 2, 5, 6, 7, 8
- [9] Y. Hacohen, E. Shechtman, and D. Lischinski. Deblurring by Example Using Dense Correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013. 1, 7
- [10] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf. Multi-frame blind deconvolution, super-resolution, and saturation correction via incremental EM. In *Proceedings of the IEEE International Conference on Computational Photography*, 2010. 1
- [11] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011. 1
- [12] Z. Hu, S. Cho, J. Wang, and M.-H. Yang. Deblurring Low-light Images with Light Streaks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1
- [13] J. Jancsary, S. Nowozin, and C. Rother. Loss-specific training of non-parametric image restoration models: A new state of the art. In *Proceedings of the European Conference on Computer Vision*, 2012. 4
- [14] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF Estimation using Sharp Edge Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1
- [15] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman. Image deblurring and denoising using color priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1
- [16] R. Kohler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and Playback of Camera Shake: Benchmarking Blind Deconvolution with a Real-world Database. In *Proceedings of the European Conference on Computer Vision*, 2012. 1, 5, 7
- [17] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*. 2009. 3, 6
- [18] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1
- [19] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding blind deconvolution algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2354–2367, Dec. 2011. 1, 3, 5, 6
- [20] W. Li, J. Zhang, and Q. Dai. Exploring Aligned Complementary Image Pair for Blind Motion Deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1
- [21] Y. Liu, J. Wang, S. Cho, A. Finkelstein, and S. Rusinkiewicz. A no-reference metric for evaluating the quality of motion deblurring. *ACM Trans. Graph.*, 32(6):175:1–175:12, Nov. 2013. 1
- [22] T. Michaeli and M. Irani. Blind Deblurring Using Internal Patch Recurrence. In *Proceedings of the European Conference on Computer Vision*, 2014. 1
- [23] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1, 5, 6, 7
- [24] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3):795–804, July 2006. 1
- [25] C. Rother. Regression tree fields - an efficient, non-parametric approach to image labeling problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 4
- [26] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative Non-blind Deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1, 4
- [27] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf. A Machine Learning Approach for Non-blind Image Deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1
- [28] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3):73:1–73:10, Aug. 2008. 1, 2, 5, 6, 7, 8
- [29] F. Sroubek and P. Milanfar. Robust multichannel blind deconvolution via fast alternating minimization. *Trans. Img. Proc.*, 21(4):1687–1700, Apr. 2012. 1

- [30] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based Blur Kernel Estimation Using Patch Priors. In *Proceedings of the IEEE International Conference on Computational Photography*, 2013. 1, 5, 7
- [31] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform Deblurring for Shaken Images. *Int. J. Comput. Vision*, 98(2):168–186, June 2012. 1
- [32] L. Xu and J. Jia. Two-phase Kernel Estimation for Robust Motion Deblurring. In *Proceedings of the European Conference on Computer Vision*, Berlin, Heidelberg, 2010. 1, 2, 5, 6, 7, 8
- [33] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. volume 26, July 2007. 1
- [34] H. Zhang and L. Carin. Multi-Shot Imaging: Joint Alignment, Deblurring and Resolution-Enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1
- [35] H. Zhang, D. Wipf, and Y. Zhang. Multi-observation blind deconvolution with an adaptive sparse prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1628–1643, Aug. 2014. 1, 5, 6
- [36] S. Zheng, L. Xu, and J. Jia. Forward Motion Deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1465–1472, Dec. 2013. 1
- [37] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling Noise in Single Image Deblurring Using Directional Filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1
- [38] Y. Zhou and N. Komodakis. A MAP-estimation Framework for Blind Deblurring Using High-level Edge Priors. In *Proceedings of the European Conference on Computer Vision*, 2014. 1
- [39] X. Zhu, F. Sroubek, and P. Milanfar. Deconvolving PSFs for a Better Motion Deblurring Using Multiple Images. In *Proceedings of the European Conference on Computer Vision*, 2012. 1