

# Learning to Estimate and Remove Non-uniform Image Blur

Florent Couzinié-Devy<sup>1,\*</sup> Jian Sun<sup>3,2,\*</sup> Karteek Alahari<sup>2,\*</sup> Jean Ponce<sup>1,\*</sup>

<sup>1</sup>École Normale Supérieure <sup>2</sup>INRIA <sup>3</sup>Xi'an Jiaotong University

## Abstract

This paper addresses the problem of restoring images subjected to unknown and spatially varying blur caused by defocus or linear (say, horizontal) motion. The estimation of the global (non-uniform) image blur is cast as a multi-label energy minimization problem. The energy is the sum of unary terms corresponding to learned local blur estimators, and binary ones corresponding to blur smoothness. Its global minimum is found using Ishikawa's method by exploiting the natural order of discretized blur values for linear motions and defocus. Once the blur has been estimated, the image is restored using a robust (non-uniform) deblurring algorithm based on sparse regularization with global image statistics. The proposed algorithm outputs both a segmentation of the image into uniform-blur layers and an estimate of the corresponding sharp image. We present qualitative results on real images, and use synthetic data to quantitatively compare our approach to the publicly available implementation of Chakrabarti *et al.* [5].

## 1. Introduction

Many photos are corrupted by camera shake, moving objects, and out-of-focus areas. This is as true for personal snapshots as for professional pictures in newspapers, fashion magazines, or scientific articles. Short exposures and small apertures can be used to limit motion blur and increase depth of field, but this may result in noisy images, especially under low light conditions. It is therefore desirable to model the blurring process, and use the image content itself to estimate the corresponding parameters and restore a sharp image. This problem is known as *blind deblurring* (or *blind deconvolution*), and it is the topic of this presentation.<sup>1</sup>

We limit our attention to defocus and linear (say, hori-



Figure 1. Two images demonstrating defocus and motion blur, with an out-of-focus swan in the foreground (left), and a moving bus before a static background (right) respectively.

zontal) motion blur. Although this setting excludes general camera shake (that can often be modeled as a pure rotation [35]) or curvilinear object motions, it is of considerable interest in many practical applications, including sports and macro photography. It is also quite challenging, since, as demonstrated by Figure 1, out-of-focus regions and relative object motions cause spatially-varying levels of blur. To account for these effects, we propose to decompose the deblurring process into two steps: (1) estimating the non-uniform blur kernel by combining learned local blur evidence with global smoothness constraints, then (2) recovering the sharp image using a robust deconvolution algorithm based on sparse regularization with global image statistics.

### 1.1. Related Work

There have been many attempts in the past to solve the image deblurring problem. Amongst these, it is commonly assumed that the blur kernel is spatially uniform [4, 7, 15, 18, 26, 37, 38], which allows it to be estimated from global image evidence. Levin *et al.* [27] argue that it is desirable to first estimate the blur kernel before using it to deblur the image. Fergus *et al.* [13] propose a Bayesian framework for the kernel estimation task using a variational optimization method. Statistical gradient priors [33], sharp edge assumptions [22, 36], and non-convex regularization [24] have also been imposed on the latent sharp image for blur estimation. Although these approaches may give impressive results, they assume that the blur kernel is uniform which, as demonstrated by Figure 1, is not realistic in many settings involving out-of-focus regions or blur due to moving objects.

\*WILLOW project-team, Département d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

<sup>1</sup>In contrast, *non-blind deblurring* refers to the simpler (but still quite challenging) problem of recovering the sharp image when the blur parameters are known.

The uniform kernel assumption has recently been relaxed in several blind deblurring methods that assume instead that blur is mostly due to camera rotation, which is realistic for camera shake in long exposures [6, 8, 16, 17, 21, 35]. In this case, the blurry image can be seen as an integral over time of images related to each other by homographies [34, 35]. An effective framework has also been proposed in [17] to approximate the spatially-varying blur kernels by combining a set of localized uniform blur kernels. Such works handle a specific type of non-uniform blur, where a global camera motion constraint can be imposed over the kernels, which simplifies the problem of kernel estimation. In contrast, our method is applicable to more general non-uniform blurs, where such constraints are not applicable.

## 1.2. Proposed Approach

We propose a method for joint image segmentation and deblurring under defocus and linear (say, horizontal) motion blur. Our approach is related to previous methods, such as [5, 9, 25, 28], but with significant differences and advantages. Liu *et al.* [28] detect blurry regions, but do not estimate the exact kernels that affect them. The algorithms proposed in [1, 8] rely on multiple blurry images or video frames to reduce the ambiguity of motion blur estimation and segmentation. Chakrabarti *et al.* [5] show interesting results for separating the blur and sharp regions in an image, but do not address deblurring itself, which is equally challenging. Dai and Wu [9] and Levin [25] rely on different local spectral or gradient cues, as well as natural image statistics for motion blur estimation. It is not clear whether these methods easily extend to defocus blur. In summary, previous approaches to our deblurring problem either (i) fall short in the estimation or the deblurring step, (ii) require multiple images, or (iii) consider a limited set of blur types (e.g., uniform, constant velocity motion, camera rotation).

We aim to overcome these limitations, and cast the estimation of the global (non-uniform) image blur as a multi-label energy minimization problem (Section 2). The energy is the sum of unary terms corresponding to learned local blur kernel estimators (Section 2.1), and binary ones corresponding to blur smoothness. Its global minimum is found using Ishikawa’s method by exploiting the natural order of discretized blur values for linear motions and defocus. Once the blur has been estimated, the image is restored using a robust (non-uniform) deblurring algorithm based on sparse regularization with global image statistics (Section 3). The proposed algorithm outputs both a segmentation of the image into uniform-blur layers and an estimate of the corresponding sharp image. We present qualitative results on real images, and use synthetic data to quantitatively compare our approach to the publicly available implementation of Chakrabarti *et al.* [5] (Section 4).

## 2. Estimating the Image Blur

We show in this section that estimating the non-uniform blur of an image can be cast as a segmentation problem, where uniform regions correspond to homogeneous blur strength. Local (but noisy) blur estimators are learned using logistic regression. A robust global estimate of the image blur is then obtained by combining the corresponding local estimates with smoothness constraints in a multi-label energy minimization framework, where labels correspond to integer (rounded) blur strengths. Since integer labels admit a natural order, it is then possible to find the global minimum of the energy using appropriate smoothness terms and Ishikawa’s method [10, 19].

### 2.1. Learning Local Blur Estimators

For simplicity, we model horizontal blur as a moving average, and defocus by a Gaussian filter. Both kernels can be parameterized by an integer “strength”  $\sigma$  corresponding to the filter size. Although it is a priori possible to learn a predictor for  $\sigma$  using regression, the energy minimization scheme presented in Section 2.2 requires a prediction score for each value of  $\sigma$  in a fixed interval. This suggests casting instead the prediction of  $\sigma$  as a multi-class classification problem, as explained below. Our predictors are *local*—that is, we estimate  $\sigma$  for each pixel. Training data is obtained by (globally) blurring a set of natural sharp images for each value of  $\sigma$ .

**Extracting local features.** We represent the local grey level pattern around each pixel in a blurry image by a feature vector  $x$  of dimension  $L + 1$  obtained by pooling the responses of a fixed bank of  $L$  multi-scale filters. Concretely, we record the average of the absolute value of each filter’s response in some neighborhood of the pixel (mean pooling). The resulting feature vectors are then rescaled to have unit norm.<sup>2</sup> The filters used in our framework are a combination of 64 Gabor filters and of atoms of a dictionary learned on blurry and sharp natural images since these have been shown to prove useful in many image restoration tasks [11]. The dictionary is learned such that some of its atoms represent blurry patches. This is achieved by first learning a small dictionary [11] from blurry image patches alone. We then learn the complete dictionary, where the initial atoms are fixed to those learned from blurry patches, with sharp image patches. Figure 2 (left) shows an illustration of dictionary-based filters learned for the horizontal motion blur case. Note that the atoms shown here in the top few rows correspond to blurry patches.

<sup>2</sup>Our filters are designed to give zero values on uniform patches since the overall grey level is irrelevant for blur estimation. To avoid noisy responses to near-uniform patches, we add one bin to our feature vectors (hence their  $L + 1$  dimensionality) before rescaling, and set the corresponding value to 1 for uniform patches, and to 0 for others.

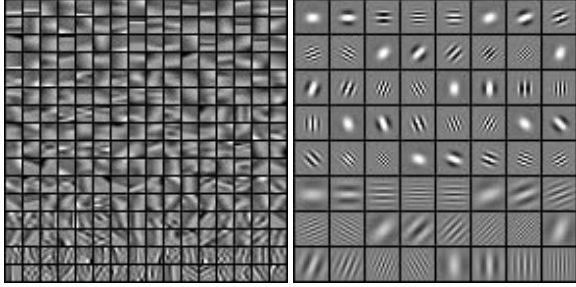


Figure 2. An illustration of filters used for generating the features. Left: a dictionary learned on blurry (horizontal motion blur in this case) and sharp natural images. Right: Gabor filter bank.

**Learning algorithm.** As mentioned earlier, we must be able to compute for each feature vector  $\mathbf{x}$  and each integer value  $k$ , associated with some interval  $I_k$  of  $\sigma$  values, a score (cost function) reflecting how (un)likely the actual value of  $\sigma$  associated with  $\mathbf{x}$  is to fall in that interval. We have chosen to use logistic regression to learn an estimator of the conditional probability  $\mathcal{P}(\sigma \in I_k | \mathbf{x})$  for each interval  $I_k$ . A suitable cost function can then be obtained by applying any decreasing function to this probability.

In practice, we divide the useful range  $[0, \Sigma]$  of blur values into  $K$  intervals  $I_k = [\sigma_{k-1}, \sigma_k]$  for  $k = 1, \dots, K$ , and model  $\mathcal{P}(\sigma > \sigma_k)$  as  $f_k(\mathbf{x}) = 1/[1 + \exp(-\mathbf{w}_k^T \mathbf{x})]$ , where  $\mathbf{w}_k$  is a vector of parameters to be learned. Given a set of training feature vectors  $\mathbf{x}_i$  with associated values  $\sigma_i$  ( $i = 1, \dots, N$ ), these parameters are learned independently for each  $k$  by using regularized logistic regression and minimizing with respect to  $\mathbf{w}_k$  the expression

$$\frac{1}{N} \sum_{i=1}^N \log[1 + \exp(-y_i \mathbf{w}_k^T \mathbf{x}_i)] + \lambda \|\mathbf{w}_k\|_2^2, \quad (1)$$

where  $y_i$  is equal to 1 if  $\sigma_i > \sigma_k$  and  $-1$  otherwise, and  $\lambda \|\mathbf{w}_k\|_2^2$  is an  $L_2$  regularization term. This convex problem is easily solved using existing optimization toolboxes such as *liblinear* [12].

An estimate for the conditional probability  $\mathcal{P}(\sigma \in I_k | \mathbf{x})$  is now easily obtained as  $f_k(\mathbf{x}) - f_{k-1}(\mathbf{x})$ ,<sup>3</sup> and the corresponding cost function can be taken to be

$$\mathbf{U}(\mathbf{x}, k) = \exp(-\gamma(f_k(\mathbf{x}) - f_{k-1}(\mathbf{x}))), \quad (2)$$

where  $\gamma$  is some positive parameter.

## 2.2. A Multi-Label Segmentation Problem

The segmentation problem has often been cast as a labeling problem, and this is the setting we use in our approach.

<sup>3</sup>The observant reader may have noticed that since the  $k$  estimators are learned independently, the function  $f_k - f_{k-1}$  is not guaranteed to be positive. We handle this problem in practice by shifting and rescaling the  $f_k$  values for each sample  $\mathbf{x}$  so as to obtain the proper ordering and a correct vector of probability values for the  $K$  classes.

Given a fixed number  $P$  of integer labels, we split the blur parameter space into  $P$  bins and predict a bin for each pixel. The bins can be built in several ways: for example, we can use bins of equal size, but they can also be built in an adaptive manner using image information. The results presented in Section 4 are obtained with uniform binning.

Once the bins are fixed, the problem becomes an ordered multi-class segmentation problem, which can be solved by minimizing with respect to  $y$  in  $\{1..P\}^N$  the energy function:

$$\sum_{i=1}^N \mathbf{U}(\mathbf{x}_i, y_i) + \lambda \sum_{i \neq j, j \in N(i)} \mathbf{B}(\mathbf{x}_i, \mathbf{x}_j, y_i, y_j, i, j), \quad (3)$$

where the vectors  $\mathbf{x}_i$  are, as before, features extracted from pixel  $i$ ,  $y_i$  is its label. The function  $\mathbf{U}$  is the unary cost of assigning label  $y_i$  to the feature  $\mathbf{x}_i$ , as derived in the previous section, and  $\mathbf{B}$  is a pairwise smoothness term that ensures that nearby pixels have consistent blur values. In practice, we use

$$\mathbf{B}(\mathbf{x}_i, \mathbf{x}_j, y_i, y_j, i, j) = \exp(-(\mu_b b_{ij} + \mu_c c_{ij})) |y_i - y_j|, \quad (4)$$

where  $b_{ij}$  is the probability of an edge being present between pixels  $i$  and  $j$  [30], and  $c_{ij}$  is a color contrast term [3] between these.

Our binary term is a convex function of the difference between the labels  $y_i$  and  $y_j$ . Although the minimization of an energy function such as that in Equation (3) is in general an NP-hard problem [2], it can be shown that it reduces to a min-cut/max-flow problem for ordered labels, such as ours [10, 19], and so its global optimum can be found by an efficient polynomial algorithm.

After obtaining the optimal kernel labels  $y_i$  for each pixel  $i$ , the local blur for the pixel will be represented by the motion or defocus kernel with corresponding value  $\sigma_i$ .

## 3. Deblurring the Image

We now address the problem of deblurring the blurry image. Given the blur kernel estimated for each pixel in the previous section, we can construct a non-uniform blur kernel matrix  $\hat{K}$ . Motivated by [20], we model the true kernel as  $K = \hat{K} + \delta_K$ , where  $\delta_K$  is a term to compensate for the errors in kernel estimation. Using this model, the blurry image  $\mathbf{f} = \hat{K}\mathbf{g} + \mathbf{u} + \mu$ , where  $\mathbf{f}$  and  $\mathbf{g}$  are the blurry and sharp images in vector form respectively, and  $\hat{K}$  denotes the spatially-varying blurring process in matrix form. The term  $\mathbf{u} = \delta_K \mathbf{g}$  is the error, and  $\mu$  is the noise.

Due to the hyper-Laplacian distribution of natural images in the gradient domain [23, 26], we impose a non-convex regularization over the image gradients. By further assuming that the error term  $\mathbf{u}$  is sparsely distributed in the



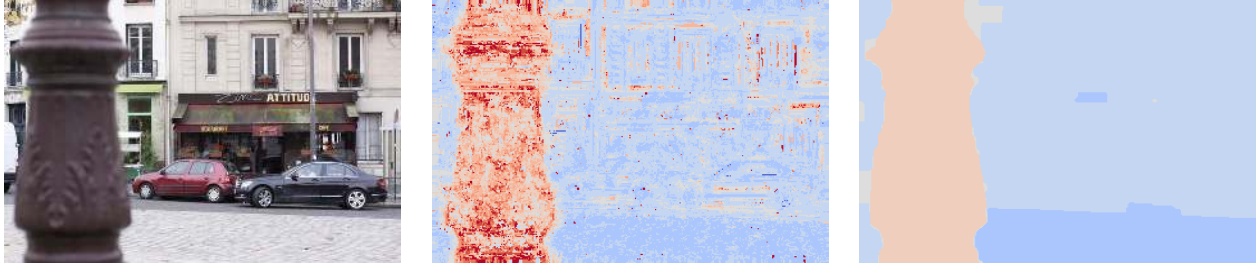


Figure 3. Result of the proposed blur estimation method on a sample image with defocus blur. From left to right: input image, estimation with only the unary cost, blur estimation with unary and binary costs, which corresponds to the global minimum of the energy function (3).

image domain, we estimate the sharp image  $\mathbf{g}$  by optimizing:

$$\min_{\mathbf{g}, \mathbf{u}} \frac{1}{2} \|\hat{K}\mathbf{g} + \mathbf{u} - \mathbf{f}\|_2^2 + \lambda_1 (\|F_1\mathbf{g}\|_\alpha^\alpha + \|F_2\mathbf{g}\|_\alpha^\alpha) + \lambda_2 \|\mathbf{u}\|_1, \quad (5)$$

where  $\|F_i\mathbf{g}\|_\alpha^\alpha = \sum_j |(F_i\mathbf{g})_j|^\alpha$  ( $\alpha$  is set to 0.8 [26]),  $F_1, F_2$  are the matrices constructed by horizontal and vertical gradient filters, and  $\lambda_1$  and  $\lambda_2$  are regularization parameters.

This non-convex energy function can be efficiently minimized by a half-quadratic splitting method [14, 23], which introduces two auxiliary vectors  $\mathbf{v}_i$  ( $i = 1, 2$ ) as:

$$\min_{\mathbf{g}, \mathbf{u}, \mathbf{v}_i} \frac{1}{2} \|\hat{K}\mathbf{g} + \mathbf{u} - \mathbf{f}\|_2^2 + \frac{\beta}{2} (\|F_1\mathbf{g} - \mathbf{v}_1\|_2^2 + \|F_2\mathbf{g} - \mathbf{v}_2\|_2^2) + \lambda_1 (\|\mathbf{v}_1\|_\alpha^\alpha + \|\mathbf{v}_2\|_\alpha^\alpha) + \lambda_2 \|\mathbf{u}\|_1.$$

When  $\beta \rightarrow \infty$ , the solution of the above problem approaches that of Equation (5). We optimize  $\mathbf{g}, \mathbf{u}$  and  $\mathbf{v}_i$  iteratively, with increasing  $\beta$ . In each iteration with a fixed  $\beta$ , we solve for  $\mathbf{g}, \mathbf{u}$  and  $\mathbf{v}_i$ . First,  $\mathbf{v}_i$  can be optimized by minimizing  $E(\mathbf{v}_i) = \frac{1}{2} \|\mathbf{v}_i - F_i\mathbf{g}\|_2^2 + \frac{\lambda_1}{\beta} \|\mathbf{v}_i\|_\alpha^\alpha$ , which can be efficiently done as in [23]. Second, by setting  $\frac{\partial E(\mathbf{g})}{\partial \mathbf{g}} = 0$ , the optimal  $\mathbf{g}$  can be found by solving the linear equations:

$$(\hat{K}^T K + \beta F_1^T F_1 + \beta F_2^T F_2)\mathbf{g} = \hat{K}^T(\mathbf{f} - \mathbf{u}) + \beta F_1^T \mathbf{v}_1 + \beta F_2^T \mathbf{v}_2, \quad (6)$$

which can be done by limited-memory quasi-Newton algorithm [31]. The large-scale matrix multiplications in Equation (6) can be implemented by local operations around pixels, e.g., spatially-varying convolution. Third,  $\mathbf{u}$  can be optimized by minimizing  $E(\mathbf{u}) = \frac{1}{2} \|\mathbf{u} - (\mathbf{f} - \hat{K}\mathbf{g})\|_2^2 + \lambda_2 \|\mathbf{u}\|_1$ , which can be solved by soft-thresholding.

The introduction of error term  $\mathbf{u}$  can effectively reduce ringing artifacts caused by possible kernel errors especially around the segmentation boundaries. In our implementation, we set  $\lambda_1 = 10^{-3}, \lambda_2 = 10^{-3}$ , and the weight  $\beta$  increases from  $\lambda_1$  to  $2^8 \lambda_1$ , with ratio of 2 in eight iterations during optimization.

## 4. Experiments

Obtaining a quantitative evaluation of algorithms for spatially-varying blur is a difficult task. There is no simple

	Horizontal	Gaussian
Blur only [5]	0.69	NA
With object model [5]	0.81	NA
Our results: multi-class	0.76	0.71
Our results: binary	0.79	0.73

Table 1. Quantitative evaluation of the estimated blur on the synthetic dataset. We compute the mean intersection vs. union score over all the images in the dataset for all the methods here. The results of [5] are shown for both its steps, using blur cues only (which is comparable to our method), and with blur and object cues. We also tested our multi-label framework, which handles images with multiple blur levels, in a binary setting, where there are exactly two labels – one to describe the sharp regions, and another for the blurry areas.

way of obtaining a ground truth to evaluate the blur estimation or the deblurring results. We believe that some quantitative results, even imperfect, will help compare methods and guide future research on the subject. Thus, we have built a synthetic dataset, where the region blurred and the strength of the blur are known. We present details of this dataset in Section 4.1. We evaluate our approach for non-uniform blind deblurring at several levels. First, we evaluate our energy formulation for blur prediction (Section 2.2), with and without the influence of the smoothness cost. Given this blur estimation, we then evaluate the proposed deconvolution method, and compare with the popular Richardson-Lucy algorithm [29, 32]. We also compare the quality of the resulting sharp image with that obtained from two baseline deblurring algorithms based on [13, 33].

### 4.1. Datasets

For a quantitative evaluation of our blur prediction and deblurring methods, we introduce a synthetic dataset. It consists of 4 sharp images, which were subjected to different levels of horizontal and Gaussian blurs. To achieve this, we manually selected regions of interest (e.g., an object or a part of the background), applied different types and levels of blurs, and added a Gaussian noise (of variance 1). In essence, this produces ground truth (blur estimate) segmentations and corresponding blurred images for each sharp image. The dataset consists of 56 images in all. Note

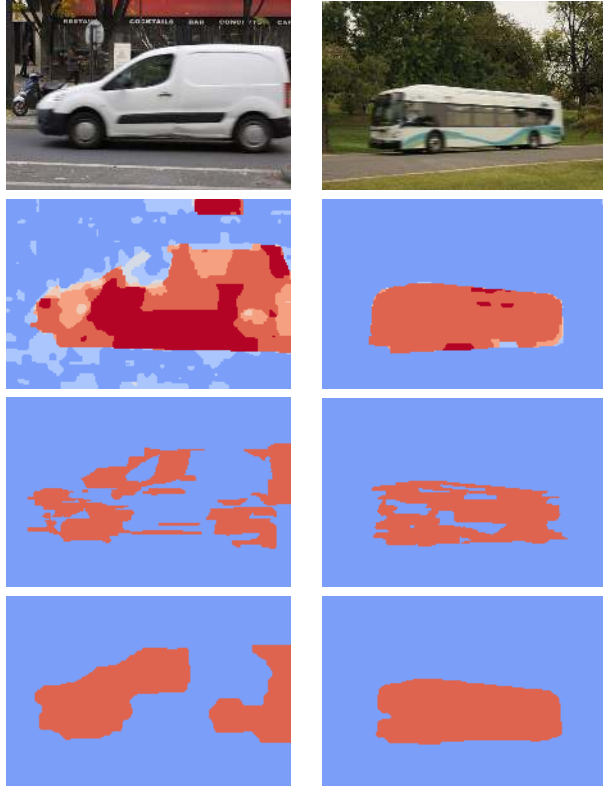


Figure 4. Blur estimation for real (left) and synthetic (right) images with horizontal motion blur. From top to bottom: input image, estimated blur (regions shown in red) with unary and binary costs, the result of [5] using only blur cues, and [5] with blur as well as object cues. We observe that our approach better handles texture-less regions.

that this synthetic dataset may not very accurate, in particular, near the blur boundaries. Nevertheless, it provides us a good test bed for evaluating different algorithms quantitatively. A few sample images from the synthetic dataset can be seen in Figures 4, 5.

We also show a qualitative evaluation of our approach on real images with different types and levels of blur, such as the 11 images used in [5], images from [25], and some of our own images. A selection of these results are shown in the paper. For the remainder of the results, the reader may refer to the project webpage.<sup>4</sup>

In our experiments, we used a bank of 64 Gabor filters, with different orientations and frequencies, and a dictionary of 320 atoms learned on a set of blurry and sharp natural images to generate the feature set for an image. The parameter  $\gamma$  in Equation (2) is set to 10, and  $\mu_b$  and  $\mu_c$  from Equation (4) are set to 25 and 200 respectively. We set the parameter that balances the relative strength of unary and binary terms,  $\lambda$  to 1.

<sup>4</sup><http://www.di.ens.fr/willow/research/nonuniblur>



Figure 5. The *car* and *horse* images from our synthetic dataset are shown here with horizontal motion blur. The *bus* image is shown in Figure 4 (right).

Image	Richardson-Lucy	Our framework
Bus	26.55	33.66
Car	27.60	37.53
Horse	24.04	34.25

Table 2. A comparison of PSNR values of our non-blind deblurring (deconvolution) method with that produced by an adapted version of the Richardson-Lucy algorithm. Here we show the average values for three images from our synthetic dataset, each subjected to six levels of horizontal blur.

## 4.2. Blur Estimation

We evaluated our local blur estimators (i.e., unary cost) on regions extracted from uniformly blurred images. In the horizontal blur case, the task is to predict one value from the set  $\sigma_i = \{1\ 3\ 5\ 7\ 9\ 11\ 13\}$ , and in the Gaussian blur case, we used a set of 9 blurs of variances uniformly spread between 0 and 4. We predicted the blur at each pixel individually, with an accuracy of 72% and 62% in the horizontal and Gaussian blurs respectively. A visualization of blur prediction on an entire image using unary costs alone, i.e.,  $\lambda = 0$  in Equation (1), is shown in Figure 3 (middle). These prediction results, although promising, are quite noisy.

We introduce the smoothness term  $\mathbf{B}$  (4) to ensure that nearby pixels have consistent blur values. The influence of this term can be seen in Figure 3 (right). We evaluated this quantitatively with the standard intersection vs. union score in Table 1. A comparison with [5] is also shown. The method by [5] uses a two-step process: (i) blur cues are first used to construct an initial blur estimate segmentation; and then (ii) a color model is learned for each region to yield the final segmentation. We show the results of these two steps (see also Figure 4). Since our approach only uses blur cues, it would be fair to compare it with the results from step (i). As seen in the figure, our segmentation results are comparable to, if not better than, the results of step (i). We observe that our method better handles texture-less regions. Note that adding object-level priors to our approach are likely to improve the results even further.

Our framework handles images with multiple blur levels (see Figure 3 (right) for example, which shows three distinct blur regions). We tested this generic framework in a binary setting, where only one blur level is assumed, with the other label corresponding to sharp regions, similar to [5]. This

Image	[13]	[33]	Our framework
Bus	28.04	24.14	31.98
Car	21.97	31.55	35.09
Horse	23.77	34.77	34.73

Table 3. PSNR values obtained with our method and two state-of-the-art uniform blind deconvolution algorithms on the *bus*, *car*, and *horse* images from our synthetic dataset (see Figures 4, 5). We provide a good approximation for uniform (horizontal) blur regions with a manually marked bounding box enclosing the blurry object. Our method, despite the lack of such ‘ground truth’ blur regions, performs better than the two other methods here.

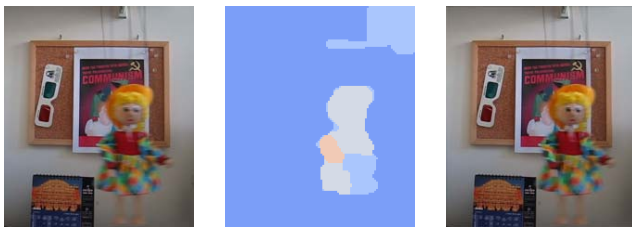


Figure 6. Segmentation and deblurring result for an image from [25]. Left to right: Blurry image, estimated segmentation, deblurred image.

2-label approach shows a performance comparable to [5] with object cues, and outperforms it when only blur cues are used.

### 4.3. Deblurring

Given the computed non-uniform blur in an image, we estimate the sharp image with our deblurring method. Since there appear to be no deconvolution methods that handle gracefully non-uniform blurs considered here, we adapted three methods to make our baseline comparisons. First, following [35], we adapted Richardson-Lucy to a known non-uniform blur. We show this comparison as average PSNR values, computed on three synthetic images with six strengths of blur each, in Table 2. We also compared our deblurring results with two state-of-the-art uniform blind deblurring algorithms [13, 33]. We applied these methods on a (manually marked) bounding box tightly enclosing the blurred object, which provides a good approximates for a uniform blur region. These results are shown in Table 3. We observe that our method, which requires no such ‘ground truth’ blur regions, outperforms [13] significantly, and is comparable to or better than [33]. In Table 4 we compare the PSNR values of blurred and estimated sharp image.

We also evaluated our blur estimation and deconvolution methods qualitatively on real images from other [5, 25], as well our own datasets. A selection of these results are shown in Figures 6, 7, and 8 for horizontal motion and defocus blurs. The interested reader is encouraged to see other, and high resolution images on the project website.

	Horizontal blur	Gaussian blur
Blurry image	33.21	33.17
Our framework	33.86	33.93

Table 4. Average PSNR values on the synthetic dataset for horizontal and Gaussian blur types. Here we show the PSNR for sharp vs. blurry images (as Blurry image), and sharp vs. deblurred images (as Our framework).

## 5. Discussion

We presented a novel approach for first estimating non-uniform blur caused by horizontal motion or defocus, and then the sharp (deconvolved) image. We demonstrated its promise through experiments with real as well as synthetic data. The quantitative evaluations on synthetic data provide a good base to compare future methods more concretely. A promising direction future work is the construction of a tree structure estimator to be able to handle two different types of blur in the same image.

**Acknowledgments.** This work was supported in part by the European Research Council (project VideoWorld), the Institut Universitaire de France, and the Quaero programme funded by the OSEO.

## References

- [1] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *ICCV*, 2007.
- [2] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete applied mathematics*, 123(1-3):155–225, 2002.
- [3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, volume 1, pages 105–112, 2001.
- [4] J. F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, pages 104–111, 2009.
- [5] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *CVPR*, pages 2512–2519, 2010.
- [6] S. Cho, H. Cho, Y. W. Tai, and S. Lee. Registration based non-uniform motion deblurring. *Computer Graphics Forum*, 31(7):2183–2192, 2012.
- [7] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics*, 28(5):145, 2009.
- [8] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007.
- [9] S. Dai and Y. Wu. Motion from blur. In *CVPR*, 2008.
- [10] J. Darbon. Global optimization for first order markov random fields with submodular priors. *Discrete Applied Mathematics*, 157(16):3412–3423, 2009.
- [11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 15(12):3736–3745, 2006.
- [12] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.





Figure 7. Horizontal blur: Sample deblurring results on two real images from [5] and on one synthetic image. From left to right: blurry image, deblurred image, close-up corresponding to the boxes shown in red. (Best viewed in pdf.)

- [13] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3):787–794, 2006.
- [14] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Processing*, 4(7):932–946, 1995.
- [15] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *ECCV*, pages 622–635, 2012.
- [16] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, pages 171–184, 2010.
- [17] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *ICCV*, pages 463–470, 2011.
- [18] Z. Hu and M.-H. Yang. Good regions to deblur. In *ECCV*, volume V, pages 59–72, 2012.
- [19] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Trans. PAMI*, 25(10):1333–1336, 2003.
- [20] H. Ji and K. Wang. Robust image deblurring with an inaccurate blur kernel. *IEEE Trans. Image Processing*, 21(4):1624–1634, 2012.
- [21] H. Ji and K. Wang. A two-stage approach to blind spatially-varying motion deblurring. In *CVPR*, pages 73–80, 2012.
- [22] N. Joshi, R. Szeliski, and D. J. Kriegman. Psf estimation using sharp edge prediction. In *CVPR*, 2008.
- [23] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, pages 1033–1041, 2009.
- [24] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, pages 233–240, 2011.
- [25] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, pages 841–848, 2007.
- [26] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 26(3):70, 2007.



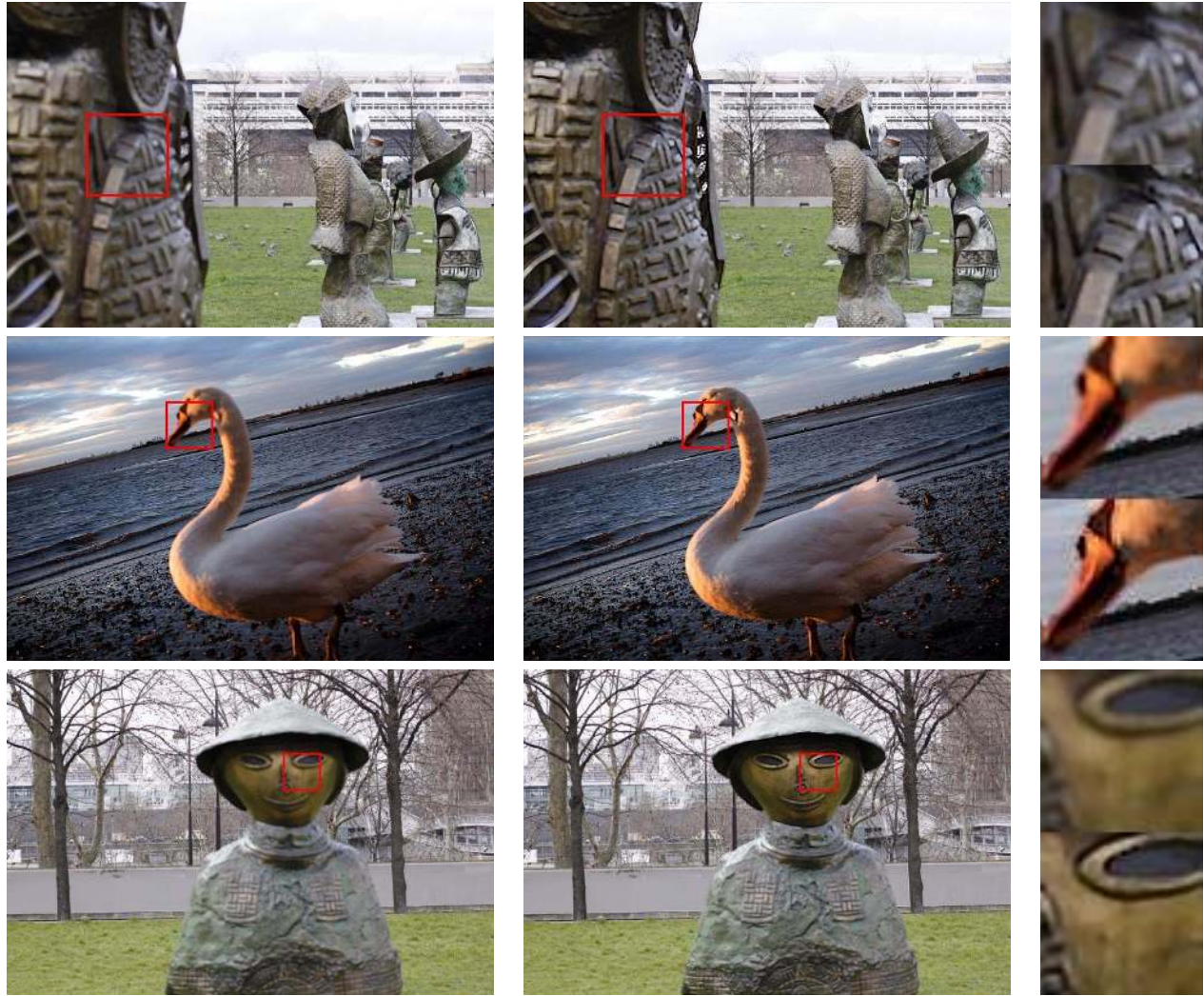


Figure 8. Defocus blur: Sample deblurring results on real images. From left to right: blurry image, deblurred image, close-up corresponding to the boxes shown in red. Note that our estimated deblurred image has more detail. (*Best viewed in pdf.*)

- [27] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971, 2009.
- [28] R. Liu, Z. Li, and J. Jia. Image partial blur detection and classification. In *CVPR*, 2008.
- [29] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79:745–754, 1974.
- [30] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26(5):530–549, 2004.
- [31] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.
- [32] W. H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.
- [33] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3):73, 2008.
- [34] Y. W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *IEEE Trans. PAMI*, 33(8):1603–1618, 2011.
- [35] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2):168–186, 2012.
- [36] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, volume I, pages 157–170, 2010.
- [37] L. Yuan, J. Sun, L. Quan, and H. Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics*, 26(3):1, 2007.
- [38] X. Zhu, F. Sroubek, and P. Milanfar. Deconvolving psfs for a better motion deblurring using multiple images. In *ECCV*, volume V, pages 636–647, 2012.