Interleaved Regression Tree Field Cascades for Blind Image Deconvolution

Kevin Schelten[1] Sebastian Nowozin[2] Jeremy Jancsary[3] Carsten Rother[4] Stefan Roth[1]
[1]TU Darmstadt [2]Microsoft Research [3]Nuance Communications [4]TU Dresden

Abstract

*Image blur from camera shake is a common cause for poor image quality in digital photography, prompting a significant recent interest in image deblurring. The vast majority of work on blind deblurring splits the problem into two subsequent steps: First, the blur process (*i.e.*, blur kernel) is estimated; then the image is restored given the estimated kernel using a non-blind deblurring algorithm. Recent work in non-blind deblurring has shown that discriminative approaches can have clear image quality and runtime benefits over typical generative formulations. In this paper, we propose a cascade for blind deblurring that alternates between kernel estimation and discriminative deblurring using regression tree fields (RTFs). We further contribute a new dataset of realistic image blur kernels from human camera shake, which we use to train the discriminative component. Extensive qualitative and quantitative experiments show a clear gain in image quality by interleaving kernel estimation and discriminative deblurring in an iterative cascade.*

1. Introduction

Camera shake causes light quantities of several, spatially distinct locations of the scene to coincide at a single coordinate of the image plane during exposure. Modern cameras stabilize the lens or the sensor, but this can only counteract relatively small camera motion. Besides limiting the user experience in consumer digital photography, image blur from camera shake is also encountered in scientific and industrial applications, causing wide interest in removing the effects of such blur [28].

The most widely adopted restoration approach is to first estimate the blur kernel [4, 7, 10, 14, 15, 30, 32, 33], often by making some statistical assumptions on the unknown sharp image. In a separate, *non-blind* step the sharp image is then restored given the kernel estimate, which is *held fixed during the procedure*. Many modern non-blind deblurring algorithms adopt a *generative* approach and impose prior knowledge on the image [13, 22, 34]. While accurate generative models exist, *e.g.*, high-order Markov random fields (MRFs) [22], their extensive computational demands prohibit the use as part of the kernel estimation phase. The origins of our work lie in recent *discriminative* approaches to non-blind deblurring, which use a neural network [23], or prediction cascades of regression tree fields (RTFs) [21] or shrinkage fields [20]. Their benefit is that they deliver high-quality image estimates, which outperform most generative approaches, at a fraction of the computational cost. However, their use in kernel estimation or blind deblurring has not been considered so far.

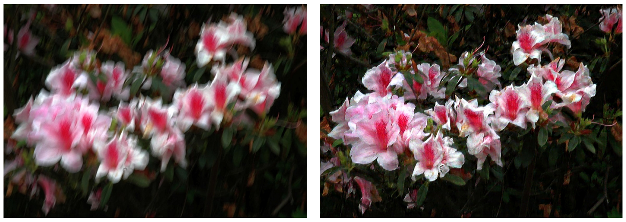

(a) Blurry (b) Deblurred

Figure 1. Deconvolution with interleaved RTF cascade.

In this paper, we propose an *RTF cascade for blind deblurring*, which alternates between discriminative deblurring and re-estimating the blur kernel using the refined image prediction. This generalizes previous work on RTF cascades for non-blind deblurring [21] to the blind deblurring task. One feature of discriminative deblurring approaches is that typical errors made by the kernel estimation procedure are learned during training, so they can be compensated for in the image recovery procedure. Kernel estimation and non-blind deblurring are therefore trained as inter-related components. To the best of our knowledge, we are the first to use discriminative image prediction custom trained to the blur updates of a blind deconvolution procedure.

To train a powerful discriminative model for image restoration, the training data should cover a rich variety of camera motions to avoid overfitting. We address this by further contributing a novel dataset of blur kernels obtained by photographing an isolated point light source under human camera shake, and use this data to train our model. We evaluate our approach extensively, and find that it clearly outperforms other recent methods from the literature.

2. Related Work

An early approach for camera shake removal was proposed by Fergus *et al*. [4], using a variational Bayesian approach. Later research [15, 31] showed that estimating the blur kernel by (approximately) marginalizing over the latent sharp image allows to cope with the ill-posed nature of the problem and also yields high-quality results in practice. Maximum a posteriori (MAP) approaches to kernel estimation typically excel in terms of fast running time [2]. However, a naïve implementation is likely to favor the trivial no-blur solution [16]. Fortunately, this can be circumvented by intermediate shock or bilateral filtering of the latent image [2, 32], alternating minimization schemes [18], or clever design of the image prior [14, 17, 24, 33].

Blind deblurring algorithms often rely on first detecting a set of useful image edges, from which the blur kernel can be estimated robustly [11, 25]. In particular, Xu and Jia [32] estimate the blur on salient edge locations and enforce the expensive kernel sparsity constraint only once, at the end of the multi-scale blur estimation procedure. Since this method yields a favorable combination of efficiency and performance (see, *e.g*., [12]), we use it to bootstrap our approach. However, note that our framework can also operate over multiple image scales, taking a delta kernel as initial input. An alternative to image-based blur estimation is to use motion sensor data recorded during exposure to reconstruct the kernel [10]. Another technique to boost restoration performance is to use context-specific sharp image examples [26]. We focus here on the more common post-capture scenario, where only the blurry image is given.

Discriminative approaches to image restoration often take the form of conditional random fields (CRFs). Due to their computational advantages, Gaussian CRFs have attracted particular attention. Tappen *et al*. [27] were among the first to propose discriminatively trained Gaussian CRFs. A more recent variant is termed regression tree fields (RTFs) [9]. The parameters of these Gaussian CRFs are determined by non-parametric regression trees; we provide more technical details on their application to image deblurring in Sec. 4. RTFs have proven effective in a variety of restoration tasks, including denoising, inpainting, and colorization [8, 9]. Recently, two different kinds of discriminative non-blind deblurring approaches have been proposed: (1) using a neural network [23], and (2) based on a stacked CRF *cascades* [20, 21]. We here rely on RTF cascades [21], since they do not require the test-time blur kernel to be known at training time, which is a prerequisite for using them as a component in a blind deblurring approach. In our work, we further explore the capacity of discriminative cascades by generalizing them to blind image deblurring through interleaving the discriminative prediction stages with blur kernel estimation.

To generalize well, a discriminative model must be exposed to a sufficiently large variety of training data. However, publicly available blurs resulting from real camera shake are limited to 8 instances in the dataset of [16], and 12 instances from the dataset of [12]. In our work, we capture realistic blurs by recording human camera shakes, and we validate the novel data by using it to train a state-of-the-art discriminative deblurring model. Note here that the recorded data will be made publicly available, and may benefit other research too, *e.g*., generative blur modeling.

3. Recording Natural Camera Shake

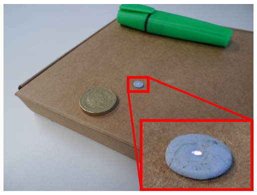


Figure 2. White LED point light source setup

Training a good regressor generally requires many instances of realistic data. To generate realistic blur kernels for training our RTF cascade, we recorded trajectories of a point light source under camera shake. For this we used a white LED (OSA Opto Light Series 400 white) as light source, placed within a cardboard box. We limited the spatial area of the light source as well as the overall amount of emitted light by placing a blue tack onto the LED, then piercing it finely with a needle, producing a white point light source of high intensity; the cardboard box is shown in Fig. 2.

We placed the box in an entirely dark room and recorded images from approximately 2 to 4 meters distance with a handheld Panasonic Lumix DMC-LX3 CCD camera. Capturing was done in 12 bit RAW format in manual mode at different focus depths (ISO 80, 500ms exposure, F2.0). We converted the images to raw TIFF using dcraw (v9.17, with `dcraw -T -v -4 -D`), then removed the constant black level. Because of the low ISO, almost no dark current noise remained in the digitized signal. Also, there were no saturated pixels. However, because of optical dispersion and different spectral sensitivities, the four color channels in the RAW frame (R, G1, G2, B) had different intensities and spatial blur. Thus, the RAW RGGB signal resembled a checkerboard pattern that could not be removed by applying a scalar gain factor to each channel. Because the green channels are the most sensitive, we simply used the G1 channel and discarded the other channels. We centered and normalized the blur kernel. Note that we did not observe any aliasing artifacts in the obtained blur kernels. Fig. 3 shows examples of recorded camera shakes. Overall, we generated 192 blur kernels. Note that this data set captures the physical process and human aspects of camera shake. Future research could involve recording spatially varying blur using a grid of LED point light sources.

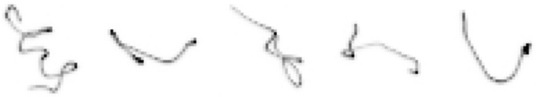


Figure 3. Instances of realistic blur kernels used for model training (Sec. 4.3). The blurs were obtained by recording the trajectory of a point light source under human camera shake.

4. Blind Deconvolution Cascades

4.1. Standard non-blind RTF cascades

As is most common, we model the formation process of image blur as convolution under additive noise, $\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n}$. Thereby, $\mathbf{y}$ denotes the blurry input image, $\mathbf{k}$ the blur kernel, $\mathbf{x}$ the unknown sharp image, and $\mathbf{n}$ the additive noise. Specifically, we follow the standard assumption of normally distributed, white noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. However, we could use a more realistic noise model such as [5] as well. Solving for the sharp image given the blur kernel is an ill-posed, difficult problem. This is partly due to sensor noise being amplified by simply inverting the kernel. Furthermore, the inverse is not properly defined if the blur kernel contains zero frequencies. Therefore, it is necessary to impose additional knowledge. In contrast to the many generative approaches to deconvolution, we here choose a recent, discriminative framework for image recovery, namely Regression Tree Fields (RTFs) [8], to model the parameters of the posterior probability $p(\mathbf{x}|\mathbf{y}, \mathbf{k})$ directly.

RTFs are Gaussian CRFs that derive their expressiveness from inferring the parameters of the local potentials using regression trees acting locally on input image features. Each tree stores at its leaves a linear term and precision matrix to define the quadratic energy contribution from the local factor variables. Regressing the potential parameters allows to overcome the apparent simplicity of Gaussian potentials, while taking full advantage of their inherent efficiency. Note that both the regression trees and the potential parameters stored at the leaves are learned in a principled, joint fashion to minimize a loss function (here, negative peak signal-to-noise ratio (PSNR)) on the training data. For more details on RTFs, we refer to [8, 9].

An RTF model for deblurring can be formulated as a Gaussian CRF of the form

$$p(\mathbf{x}|\mathbf{y}, \mathbf{k}) \propto \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}(\mathbf{y}, \mathbf{k}), \mathbf{C}(\mathbf{y}, \mathbf{k})\right), \tag{1}$$

whereby the parameters of the mean $\boldsymbol{\mu}(\mathbf{y}, \mathbf{k})$ and covariance matrix $\mathbf{C}(\mathbf{y}, \mathbf{k})$ are partly regressed from the input image $\mathbf{y}$ by the RTF framework, with the blur kernel $\mathbf{k}$ being held fixed as a constant. In more detail, let $\mathbf{T}_\mathbf{k}$ denote the Toeplitz matrix expressing convolution by blur kernel $\mathbf{k}$, such that the identity $\mathbf{T}_\mathbf{k}\mathbf{x} \equiv \mathbf{k} \otimes \mathbf{x}$ is fulfilled. Motivated by generative approaches to deblurring, Schmidt *et al.* [21] show that the covariance and mean of the Gaussian CRF in Eq. (1) may be chosen as

$$\mathbf{C}(\mathbf{y}, \mathbf{k}) = \left(\mathbf{W}(\mathbf{y}) + \frac{1}{\sigma^2}\mathbf{T}_\mathbf{k}^T\mathbf{T}_\mathbf{k}\right)^{-1} \tag{2}$$

$$\boldsymbol{\mu}(\mathbf{y}, \mathbf{k}) = \mathbf{C}(\mathbf{y}, \mathbf{k})\left(\mathbf{w}(\mathbf{y}) + \frac{1}{\sigma^2}\mathbf{T}_\mathbf{k}^T\mathbf{y}\right), \tag{3}$$

whereby the matrix $\mathbf{W}(\mathbf{y})$ and vector $\mathbf{w}(\mathbf{y})$ are regressed from the input image by the RTF framework. Overall, inference consists of regressing the CRF parameters and subsequently computing the prediction as $\operatorname{argmax}_\mathbf{x} p(\mathbf{x}|\mathbf{y}, \mathbf{k}) = \boldsymbol{\mu}(\mathbf{y}, \mathbf{k})$.

However, it is not easy to regress optimal potential parameters from the input image immediately, because the blur strongly obfuscates the image content, for example by creating ghosting-like overlays of edges in uniform regions. This can be overcome by stacking RTFs into a cascade [19], which generates a sequence of iteratively refined sharp image estimates $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$. At each level of the cascade, the corresponding RTF parameters are regressed not only from the input image, but also from the improved previous prediction, which facilitates the procedure. In particular, the matrix $\mathbf{W}_i(\mathbf{y}, \mathbf{x}_{i-1})$ and vector $\mathbf{w}_i(\mathbf{y}, \mathbf{x}_{i-1})$ are regressed at the i-th level of the cascade. Fig. 4(a) depicts a schematic illustration of the non-blind RTF cascade model from [21].

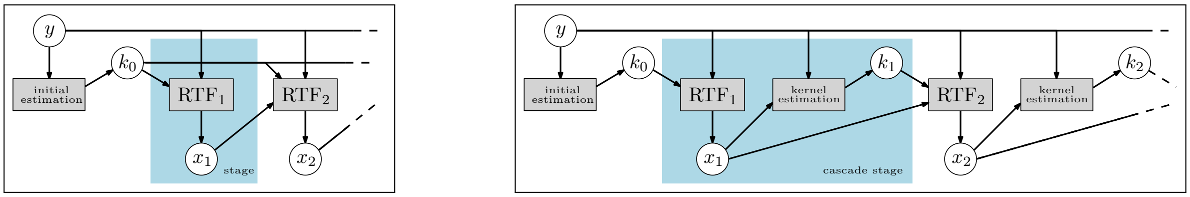
4.2. Interleaved RTF cascades

Besides their quantitative and qualitative benefits in terms of image quality and efficiency, a distinctive feature of discriminative deconvolution methods is their adaptability to kernel estimation (errors). In particular, RTF cascades yield best results when trained with blur kernels of similar kind as those provided at test time [21]. In this work, we *interleave* the image prediction steps in the RTF cascade with kernel re-estimation. Note that we here focus on uniform blur. Although the image formation model of spatially varying blur is more involved, the procedure detailed below is in principle equally valid.

We design an interleaved procedure by updating the blur kernel using the improved latent image prediction available at every cascade level, such that each RTF stage is provided with a refined kernel estimate. Note that the cascade is initialized with the kernel estimate $\mathbf{k}_0$ of an auxiliary method[1]. For higher stages $i = 1, \ldots, N$ of the interleaved cascade, the output $\mathbf{x}_i$ of the i-th RTF is used to compute a refined kernel estimate $\mathbf{k}_i$. Fig. 4(b) depicts a schematic illustration of the interleaved RTF prediction cascade.

Specifically, we compute the kernel update using the image derivatives by minimizing with respect to $\mathbf{k}$ the objec-

[1] In the experiments, we mostly use [32] to perform this step, but we can also initialize with the delta kernel when estimating the blur over the scales of an image pyramid. (In this case, the final kernel estimate at one scale is upsampled to serve as the initial estimate for the next interleaved RTF cascade, see Fig. 9.)



(a) (b)

Figure 4. Comparison of standard, [21], versus our proposed interleaved RTF cascade schemes. (a) Standard non-blind RTF cascade: The blur kernel is set to the initial blur estimate $\mathbf{k}_0$ and stays invariant over the cascade. (b) Interleaved RTF cascade: The kernel is re-estimated over cascade stages using the refined image predictions $\mathbf{x}_i$. In the experiments, we mostly use [32] to obtain the initial blur estimate, but we can also initialize with the delta kernel by using several interleaved RTF cascades to operate over the scales of an image pyramid.

```
Algorithm 1 Interleaved RTF cascade
input: Blurry image $\mathbf{y}$, initial blur kernel $\mathbf{k}_0$
output: Deblurred image $\mathbf{x}_N$, refined blur kernel $\mathbf{k}_N$

  for $i = 1, ..., N$ do
    [update latent image using $i$-th RTF regressor]
    $\mathbf{x}_i := \left(\mathbf{W}_i(\mathbf{y},\mathbf{x}_{i-1}) + \frac{1}{\sigma^2}\mathbf{T}^T_{\mathbf{k}_{i-1}}\mathbf{T}_{\mathbf{k}_{i-1}}\right)^{-1} \times$
        $\left(\mathbf{w}_i(\mathbf{y},\mathbf{x}_{i-1}) + \frac{1}{\sigma^2}\mathbf{T}^T_{\mathbf{k}_{i-1}}\mathbf{y}\right)$
    [update blur kernel]
    $\mathbf{k}_i := \mathrm{argmin}_{\mathbf{k}} \|\nabla\mathbf{y} - \mathbf{k}\otimes\nabla\mathbf{x}_i\|^2 + \gamma\|\mathbf{k}\|_1$
  end for
```

tive function

$$f(\mathbf{k}) = \|\nabla\mathbf{y} - \mathbf{k}\otimes\nabla\mathbf{x}_i\|^2 + \gamma\|\mathbf{k}\|_1. \quad (4)$$

Hereby, we let $\nabla\mathbf{x} = (\partial_1\mathbf{x}, \partial_2\mathbf{x}) = (\mathbf{f}_1\otimes\mathbf{x}, \mathbf{f}_2\otimes\mathbf{x})$ denote the canonical image gradients computed with the standard derivative filters $\mathbf{f}_1 = [1,-1]$ and $\mathbf{f}_2 = [1,-1]^T$. For the gradient image, we define convolution by the blur kernel to apply component-wise, *i.e.*, $\mathbf{k}\otimes\nabla\mathbf{x} := (\mathbf{k}\otimes\partial_1\mathbf{x}, \mathbf{k}\otimes\partial_2\mathbf{x})$. Note further that the objective for kernel re-estimation (Eq. 4) consists of a squared residuals term motivated by a Gaussian noise assumption, and an L^1-norm penalty to encourage kernel sparsity, which is weighted by a constant $\gamma > 0$. The regularization parameter γ can be learned from data (*cf*. Sec. 4.3), but even simply setting $\gamma = 1$ already yields very good results. Algorithm 1 summarizes the proposed interleaved RTF cascade for blind deblurring.

We optimize the kernel update objective $f(\mathbf{k})$ of Eq. (4) using iterative re-weighted least squares (IRLS). This means iteratively solving varying least-squares problems until the distance between consecutive solutions passes below a convergence threshold. Specifically, at the j-th iteration of IRLS, we compute

$$\mathbf{k}^j = \mathrm{argmin}_{\mathbf{k}} \|\nabla\mathbf{y} - \mathbf{k}\otimes\nabla\mathbf{x}_i\|^2 + \gamma\mathbf{k}^T\mathrm{diag}(\mathbf{z})\mathbf{k}. \quad (5)$$

Thereby, the n-th element of the weighting vector $\mathbf{z}$ is determined by $z_n = 1/\max(k_n^{j-1}, \epsilon)$, where we fixed $\epsilon = 10^{-5}$ in the experiments. Minimizing the quadratic expression in Eq. (5) is equivalent to solving a linear system of equations $\mathbf{A}\mathbf{k} = \mathbf{b}$. The left-hand side matrix of this system is determined by $\mathbf{A} = \sum_h \sum_c [\partial_h\mathbf{x}]_c [\partial_h\mathbf{x}]_c^T + \gamma\mathrm{diag}(\mathbf{z})$, using $[\cdot]_c$ to denote the c-th kernel-sized clique. On the other hand, the right-hand side vector is $\mathbf{b} = \sum_h \sum_c [\partial_h\mathbf{x}]_c (\partial_h\mathbf{y})_c$, using $(\partial_h\mathbf{y})_c$ to denote the pixel situated at the center of the c-th kernel-sized clique in the derivative image $\partial_h\mathbf{y}$. Note that this system is generally not amenable to solving by FFT. Hence we use conjugate gradients with Jacobi preconditioning, computing the diagonal of the system matrix as $\sum_h \mathbf{1}\otimes[\partial_h\mathbf{x}]^{\circ 2} + \gamma\mathbf{z}$, where $[\cdot]^{\circ 2}$ denotes component-wise Hadamard square, while $\mathbf{1}$ is an image of ones and size $\dim(\mathbf{x}) - \dim(\mathbf{k}) + [1,1]^T$.

4.3. Learning

Training data. We compiled sharp images for use as ground-truth data from two different benchmark datasets, BSDS500 [1] and PASCAL VOC [3]. Note that the training images stem from entirely different sources than those used in the experimental evaluation (Sec. 5). As blur data we used 95 realistic blur kernels generated by recording the trajectory of a light source under human camera shake (see Sec. 3). We complemented these with synthetic blurs created by projecting randomly sampled motions in 3D space onto the camera plane [21]. Note that none of these kernels is used at test time. To obtain blurry images, we synthetically convolved the ground-truth images and added Gaussian noise of standard deviation equal to 0.2% of the maximum pixel intensity. We used 336 clean and corrupted image pairs and blur kernels to train our models.

Learning the latent image prediction. At each level of the cascade, we learn a separate RTF model for image restoration. Besides the blurry input image, each model receives as additional input the previous image prediction and is further parameterized by a blur kernel of increasing refinement. This is different from [21], where the blur kernel remains fixed throughout all stages. We remark that the RTF models learned at every level adapt precisely to the kernel re-

	Fergus [4]	Cho [2]	Xu [33]	Levin [15]
∅ PSNR	29.38	29.71	29.74	30.05

Standard RTF	Interleaved RTF
31.16	**31.50**

Table 1. Average PSNR (dB) values on the test set of Levin *et al.* [15].

	Xu [32]	Cho [2]	Whyte [29]	Hirsch [7]	Krishnan [14]
∅ PSNR	29.54	28.98	28.07	27.77	25.73

Std. RTF	Interl. RTF
29.91	**30.11**

Table 2. Average PSNR (dB) values on the test set of Köhler *et al.* [12].

estimation and to the preceding image predictions given as inputs (see Tab. 3). The resulting, interleaved cascade thus forms a unit of inter-related components and needs to be trained together. We opt for regression trees of depth 7. To leverage more discriminative features than simple pixel intensities, we rely on the FoE filter bank of [6], *i.e.*, each model receives as additional features the filter responses of the previous prediction. Per depth level, we use 40 iterations of LBFGS to optimize the model parameters, with another 100 clean-up cycles after splitting the leaves at the final level 7. To accelerate the learning procedure, we did not use the original size images, but 125×125 sized pairs of degraded and sharp crops. Learning a cascade of depth 3 (plus evaluating the full interleaved model on the training images for each additional level) took 10 days on a machine with a 3.20GHz Intel Core i7 3930K CPU. Training time could be reduced by parallel computing on several machines.

Learning the blur kernel update. With regard to updating the blur kernel, the regularization parameter γ weighting the influence of likelihood and prior in the objective function for the kernel update (Eq. 4) may also be learned from data in a loss-based fashion. We opt for a blur kernel loss function based on the outlier resistant L1 metric, namely $\epsilon(\mathbf{k}, \mathbf{k}_{gt}) = \|\mathbf{k} - \mathbf{k}_{gt}\|_1 / |\mathbf{k}|$, where $|\mathbf{k}|$ denotes the number of kernel elements. Note here that care must be taken to align the blur kernels with each other before evaluating the distance, since a translation in the kernel simply leads to a translation in the deblurred image, and this should not be penalized. Although we could also optimize w.r.t. image quality, it is more efficient to compute the kernel loss, which obviates the more expensive image prediction step. Hence at the i-th level of the interleaved cascade, a weight γ_i can be learned to optimize the empirical risk $\langle \epsilon(\mathbf{k}_i, \mathbf{k}_{gt}) \rangle = \frac{1}{N} \sum_n \epsilon(\mathbf{k}_i^n, \mathbf{k}_{gt}^n)$ over the training data. Since this is a unimodal objective function, a simple 1D line search suffices to find the optimum.

5. Experiments

Tab. 1 gives the performance of the proposed, interleaved RTF cascade on the benchmark of Levin *et al.* [15]. Our interleaved algorithm outperforms the blind deblurring methods [2, 4, 15, 33] on this benchmark with a very large margin of at least 1.45 dB. We further evaluated the non-blind, standard RTF cascade on this benchmark, using the blur estimate of [32] as input. This guarantees a fair comparison to the interleaved RTF cascade, which, although bootstrapped with [32], re-estimates the blur iteratively over the prediction stages. We remark that standard RTF cascades are state-of-the-art in non-blind deblurring and outperform many existing sparsity-based methods [21]. Tab. 1 shows that our interleaved RTF cascade achieves significantly better results than the state-of-the-art non-blind cascade [21] by 0.34 dB in PSNR. This demonstrates how useful it is to re-estimate the blur kernel between discriminative image updates in a learned cascade.

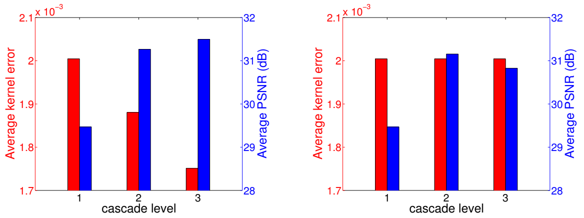
Tab. 2 gives results on the benchmark of Köhler *et al.* [12]. Here, our interleaved algorithm achieves substantially better results than a multitude of other methods [2, 7, 14, 29, 32] by at least 0.57 dB. Note that several images of the dataset [12] are very challenging, having spatially varying blur of over 100 pixels. The interleaved algorithm again outperforms its standard, non-blind counterpart by a significant margin of 0.2 dB.

Figs. 1, 6 and 8 show that our method preserves challenging regions of image texture faithfully, while suppressing ringing and noise artifacts in smooth regions or on the image boundary. Notably, Fig. 6 demonstrates visibly superior performance of the interleaved cascade over a wide variety of blind deconvolution methods, while Fig. 8 shows that interleaving with kernel updates yields a noticeably higher degree of realism in the deblurred image than using the standard cascade.

We further analyze the benefit of custom, discriminative training of the interleaved cascade to the refined kernel estimates available at each stage. Tab. 3 gives results for prediction with and without interleaved kernel updates, using RTF cascades learned with and without interleaved kernel updates. We observe that it is important to train the image prediction step based on the refined blur estimates to

	Ilvd. prediction	Std. prediction
Ilvd. training	**31.50**	30.67
Std. training	30.81	**31.16**

Table 3. Performance of RTF cascade models in average PSNR (dB) on the test set of [15]. Training and prediction is performed with interleaved ("Ilvd.") or without ("Std.") re-estimation of the blur kernel over cascade levels.



(a) Interleaved RTF cascade (b) Standard RTF cascade

Figure 5. Average blur kernel error versus image quality over interleaved and standard RTF cascade levels on the test set of Levin *et al.* [15]. The kernel error is quantified in mean absolute distance $\epsilon(\mathbf{k}, \mathbf{k}_{gt}) = \|\mathbf{k} - \mathbf{k}_{gt}\|_1 / |\mathbf{k}|$ to the ground truth blur (letting $|\mathbf{k}|$ denote the number of kernel elements). The interleaved RTF cascade simultaneously enhances the image and blur kernel.

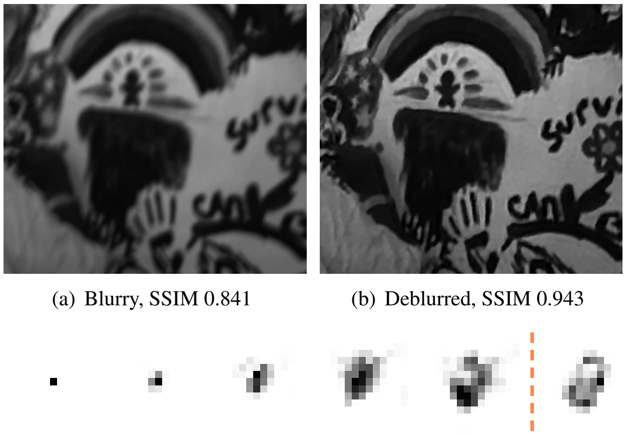
unlock the full potential of our approach. Simply interleaving a pre-trained standard cascade with blur updates leads to substantially inferior results. Note further that learning the image restoration steps expressly to extract maximum effect from the refined kernel estimates is a key benefit of discriminative updates.

To gain more insight into the role of kernel refinement over cascade stages, we rely on the dataset of Levin *et al.* [15], since it includes ground truth blur kernels to evaluate with. In particular, we measure the mean absolute distance of the (aligned) blur estimates to the ground truth kernels. Fig. 5 depicts the average kernel error versus the average image quality over all 32 image and kernel pairs of the benchmark, shown after each of three cascade levels. We observe that the increasing image quality over the cascade allows to improve the kernel estimate and vice versa, while on the other hand, holding the blur fixed over the cascade leads to inferior overall performance.

We further examine the blur refinement effect of our algorithm in a visual study, relying once more on the 8 ground-truth camera shakes of [15]. Fig 7 shows three versions of each camera movement: The kernel estimate of [32] used as initialization to the interleaved restoration process, the blur estimate from the last stage of the interleaved cascade, and the ground-truth kernel provided with the benchmark. We observe that the interleaved estimation procedure substantially enhances the initial estimate.

To measure running times, we used a 3.20GHz Intel Core i7 3930K processor. For a kernel size of 41×41, blind deconvolution with our interleaved cascade algorithm needed 98.66s for an image of size 800×800. For comparison, we measured 156.49s for the efficient deblurring algorithm of Krishnan *et al.* [14]. Note that as a prototype, our implementation is not optimized for fast running time.

Finally, to demonstrate that our approach does not require a specific auxiliary method for initialization, Fig. 9 shows an instance of multiscale interleaved RTF regression with delta kernel initialization. Hereby, interleaved cascades are used to predict image and kernel estimates at each level of a Gaussian pyramid. The estimates at one level are enlarged to serve as inputs for the next finer level. Note that the model trained with initial blur estimation from another method [32] cannot be used with delta blur initialization. Instead, going from coarse to fine, we trained progressively more powerful interleaved models to account for the higher level of image details.



(a) Blurry, SSIM 0.841 (b) Deblurred, SSIM 0.943

(c) Kernel estimates over scale. Rightmost: Ground-truth blur.

Figure 9. Multiscale interleaved RTF regression with *delta kernel initialization*. Each level of the pyramid is equipped with a progressively more powerful interleaved RTF cascade.

6. Conclusion

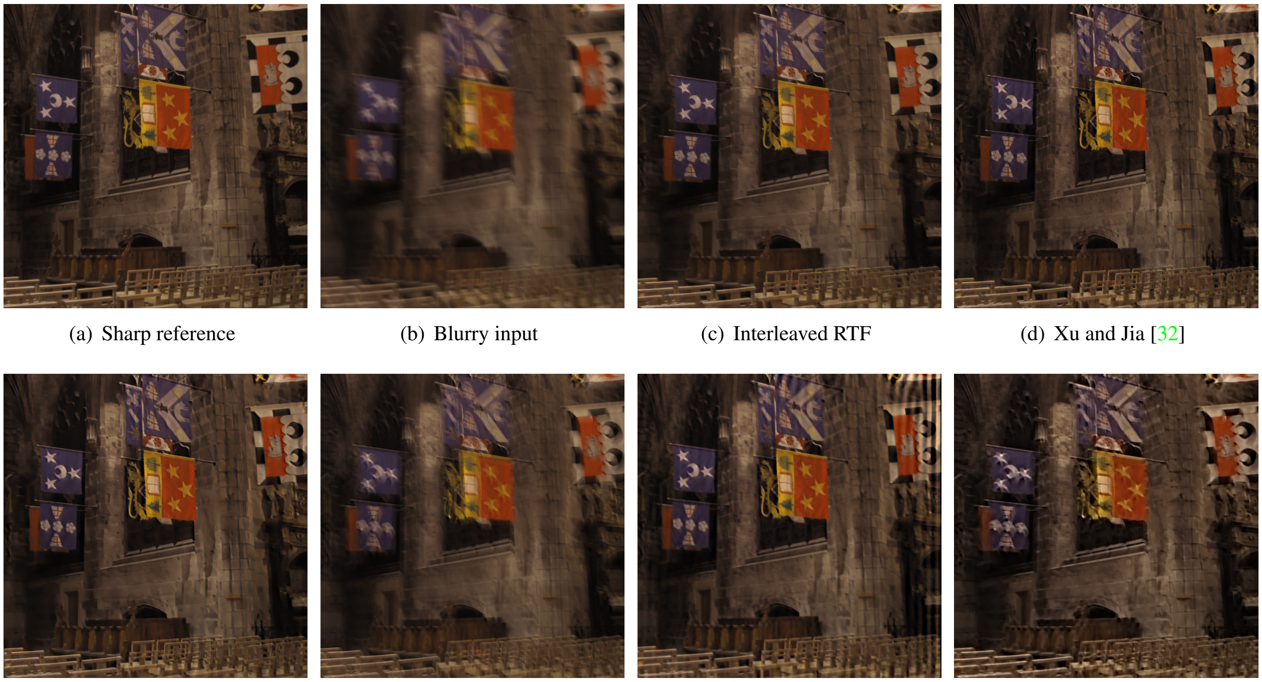
In this paper, we put forth a novel, interleaved RTF cascade model for blind deblurring that consolidates discriminative image prediction with blur estimation, whereby each step is trained expressly to fit to the other. The model is validated by extensive experimentation. Further, we contributed a novel dataset of human camera shakes by recording LED trajectories with a handheld camera, using this data to train our model. All code, data, and trained models will be made freely available.

Acknowledgments. We would like to thank Andreas Georgiou for assembling the LED point light source device. Funding is partly provided by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307942. Parts of this work have been done while KS, CR, and JJ were with Microsoft Research Ltd.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, May 2011.

[2] S. Cho and S. Lee. Fast motion deblurring. *ACM T. Graphics*, 28(5), Dec. 2009.



(a) Sharp reference

(b) Blurry input

(c) Interleaved RTF

(d) Xu and Jia [32]

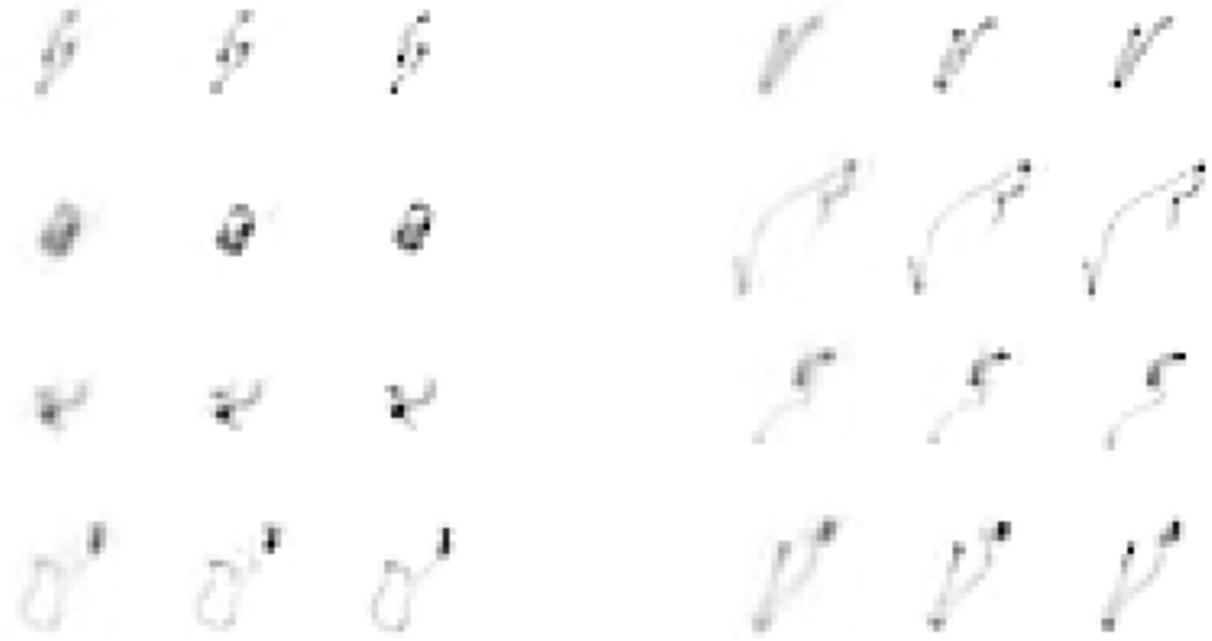
(e) Cho and Lee [2]

(f) Whyte *et al*. [29]

(g) Hirsch *et al*. [7]

(h) Krishnan [14]

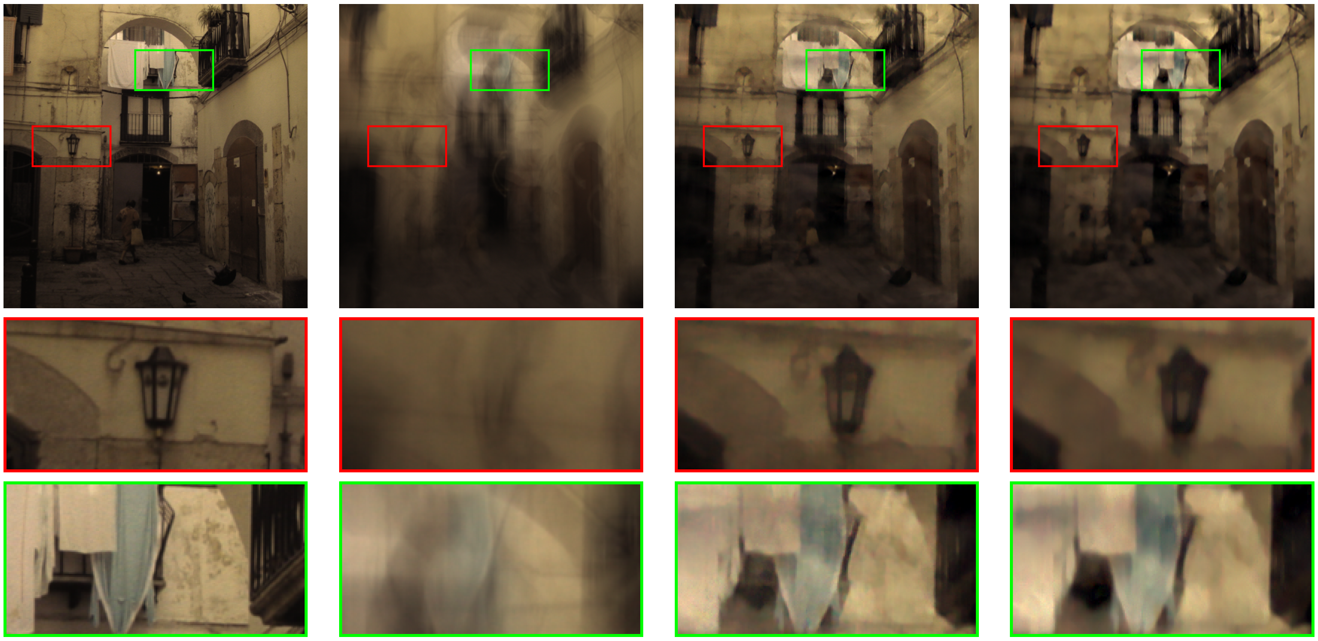
Figure 6. Qualitative comparison of deblurring algorithms on the benchmark of Köhler *et al*. [12]. The reference image shown in (a) is the first frame of the recorded motion. The interleaved RTF simultaneously recovers sharp edges such as the patterns of the flags, while keeping boundary artifacts at a minimum.



(1) Xu & Jia [30] (2) proposed (3) ground truth

(1) Xu & Jia [30] (2) proposed (3) ground truth

Figure 7. Kernel refinement on the dataset of Levin *et al*. [16]. For all of the 8 blurs in the test set, a triple is displayed horizontally. From left to right, each triple consists of: (1) The estimate of Xu and Jia [32] used to initialize the interleaved cascade, (2) the refined blur estimate at the final level of the cascade, (3) the ground-truth kernel. Each triple is scaled jointly to the full intensity range.



(a) Sharp reference (b) Blurry (c) Interleaved (d) Standard

Figure 8. Qualitative comparison of interleaved versus standard RTF cascade. The interleaved RTF cascade recovers a higher level of image details and yields a more realistic deblurring result.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM T. Graphics*, 3(25):787–794, July 2006.

[5] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE TIP*, 17(10):1737–1754, Oct. 2008.

[6] Q. Gao and S. Roth. How well do filter-based MRFs model natural images? In *Pattern Recognition (DAGM) 2012*.

[7] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *ICCV 2011*.

[8] J. Jancsary, S. Nowozin, and C. Rother. Loss-specific training of non-parametric image restoration models: A new state of the art. In *ECCV 2012*.

[9] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression tree fields — an efficient, non-parametric approach to image labeling problems. In *CVPR 2012*.

[10] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM T. Graphics*, 29(4):30:1–30.9, July 2010.

[11] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. In *CVPR 2008*.

[12] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: benchmarking blind deconvolution with a real-world database. In *ECCV 2012*.

[13] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-Laplacian priors. In *NIPS*2009*.

[14] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR 2011*.

[15] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR 2011*.

[16] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR 2009*.

[17] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV 2014*.

[18] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR 2014*.

[19] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother. Cascades of regression tree fields for image restoration. *arXiv:1404.2086*, 2014.

[20] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR 2014*.

[21] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative non-blind deblurring. In *CVPR 2013*.

[22] U. Schmidt, K. Schelten, and S. Roth. Bayesian deblurring with integrated noise estimation. In *CVPR 2011*.

[23] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf. A machine learning approach for non-blind image deconvolution. In *CVPR 2013*.

[24] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM T. Graphics*, 27(3), Aug. 2008.

[25] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP 2013*.

[26] L. Sun, S. Cho, J. Wang, and J. Hays. Good image priors for non-blind deconvolution: Generic vs specific. In *ECCV 2014*.

[27] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. Learning Gaussian conditional random fields for low-level vision. In *CVPR 2007*.

[28] R. Wang and D. Tao. Recent progress in image deblurring. *arXiv preprint arXiv:1409.6838*, 2014.

[29] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. In *IEEE Color and Photometry in Computer Vision Workshop*, 2011.

[30] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR 2010*.

[31] D. Wipf and H. Zhang. Analysis of Bayesian blind deconvolution. In *EMMCVPR 2013*.

[32] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV 2010*.

[33] L. Xu, S. Zheng, and J. Jia. Unnatural L_0 sparse representation for natural image deblurring. In *CVPR 2013*.

[34] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV 2011*.