Discriminative Non-blind Deblurring

Uwe Schmidt[⋆]   Carsten Rother[†]   Sebastian Nowozin[†]   Jeremy Jancsary[†]   Stefan Roth[⋆]
⋆ Department of Computer Science, TU Darmstadt   † Microsoft Research Cambridge

Abstract

Non-blind deblurring is an integral component of blind approaches for removing image blur due to camera shake. Even though learning-based deblurring methods exist, they have been limited to the generative case and are computationally expensive. To this date, manually-defined models are thus most widely used, though limiting the attained restoration quality. We address this gap by proposing a discriminative approach for non-blind deblurring. One key challenge is that the blur kernel in use at test time is not known in advance. To address this, we analyze existing approaches that use half-quadratic regularization. From this analysis, we derive a discriminative model cascade for image deblurring. Our cascade model consists of a Gaussian CRF at each stage, based on the recently introduced regression tree fields. We train our model by loss minimization and use synthetically generated blur kernels to generate training data. Our experiments show that the proposed approach is efficient and yields state-of-the-art restoration quality on images corrupted with synthetic and real blur.

1. Introduction

Image blur (*e.g.*, camera shake) is one of the main sources of image corruption in digital photography and hard to undo. Image deblurring has thus been an active area of research, starting with the pioneering work of Lucy [21] and Richardson [23]. Recent work has predominantly focused on *blind deblurring*, particularly on estimating the blur from images (stationary and non-stationary). However, relatively little attention has been paid to *non-blind deblurring*, that is, restoring the image given known or estimated image blur. Yet, this is an important problem since most blind deblurring approaches separate the problem into blur estimation and non-blind deblurring (theoretically justified by Levin *et al.* [20]). To this date, most approaches rely on the classical Lucy-Richardson algorithm as non-blind deblurring component [*e.g.*, 8], or use manually-defined image priors formulated as Markov random fields (MRFs) with sparse, *i.e.* non-Gaussian, potential functions [17, 18, 30]. Learning-based approaches have been restricted to gener-

atively trained models [25], but have found limited adoption due to computational challenges in inference. This is in contrast to image denoising, where discriminative approaches have been used extensively [2, 4, 14, 27], and are often characterized by state-of-the-art restoration performance combined with low computational effort.

In this paper we introduce a *discriminative* non-blind image deblurring approach for arbitrary photographic input images and *arbitrary blurs*. To the best of our knowledge, this is the first time discriminative deblurring has been attempted. Our model is discriminatively trained by minimizing an application-specific loss function (here, PSNR) on a training set. In this paper we assume stationary image blur, *i.e.* the observed image is the result of convolving the unknown original image with a blur kernel (+ noise), but our approach is not limited to this setup and can be extended to non-uniform image blurs.

In a discriminative approach, a number of challenges have to be overcome. Since it is in general not feasible to train a specialized model for every image blur, it is necessary to train a model that outputs a deblurred image given an arbitrary input image and blur kernel. We address this by effectively parametrizing our discriminative model with the blur kernel. The difficulty here is that the image distortion is only known at test time. We address this using a model cascade based on regression tree fields [15], which first predicts a relatively crude estimate that removes dominant image blur and is refined further in later stages. Moreover, sufficient training data must be available, but realistic image blurs are scarce [16, 20]. We use synthetically generated blur kernels to overcome this limitation.

We further make the following contributions: *(1)* We analyze commonly used half-quadratic regularization [11, 12] with sparse image priors, and show how our formulation naturally arises as a generalization; *(2)* we train our model with realistic, but synthetically generated blur kernels and experimentally demonstrate that a model trained in such a way generalizes to unseen real blur kernels; *(3)* we demonstrate state-of-the-art performance on a synthetically blurred test set [25] and on two realistic data sets for camera shake [16, 19]. While previous non-blind deblurring approaches have for the most part either been very fast but

with inferior performance, or slow but with high-quality results [*e.g.* 25], our approach delivers state-of-the-art deblurring performance with an efficient inference method that allows deblurring even higher-resolution images.

2. Generalizing Half-quadratic Regularization

To motivate our discriminative approach and understand its connections to the existing literature, it is beneficial to recall half-quadratic regularizers [5, 11, 12] and their relation to recent image restoration approaches[1]. In image deblurring, denoising and other restoration applications, sparse image priors are frequently used for regularization [*e.g.* 17, 18, 24]. Typically, they model an image $\mathbf{x}$ through the response of linear filters $\mathbf{f}_j$ (*e.g.*, horizontal and vertical image derivatives), which induce overlapping cliques $c \in \mathcal{C}_j$ in the corresponding Markov random field (MRF) prior:

$$p(\mathbf{x}) \propto \prod_j \prod_{c \in \mathcal{C}_j} \rho_j(\mathbf{f}_j^{\mathsf{T}} \mathbf{x}_{(c)}). \tag{1}$$

A sparse (non-Gaussian) potential function ρ_j models the filter response of $\mathbf{f}_j$ to the clique pixels $\mathbf{x}_{(c)}$.

The image corruption process is often modeled by specifying a Gaussian likelihood $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{Kx}, \sigma^2 \mathbf{I})$ for the observed, corrupted image $\mathbf{y}$. In the case of non-blind deconvolution, we have $\mathbf{Kx} = \mathbf{k} \otimes \mathbf{x}$, where $\mathbf{K}$ is the blur matrix that corresponds to convolving the image with a blur kernel $\mathbf{k}$. The image noise is assumed to be pixel-independent additive white Gaussian noise with variance σ^2. Using Bayes' theorem, one obtains the posterior distribution over the restored image as $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x})$.

The principle of half-quadratic regularization [5, 11, 12] is to ease inference (*e.g.*, MAP estimation) by introducing (independent) auxiliary/latent variables z_{jc} for each filter and image clique, such that the prior is retained by performing an operation $\bigoplus \in \{\max, \sup, \sum, \int, \ldots\}$ that eliminates the auxiliary variables:

$$\begin{aligned} p(\mathbf{x}) &\propto \bigoplus_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \\ &\propto \prod_j \prod_{c \in \mathcal{C}_j} \bigoplus_{z_{jc}} \phi_j(\mathbf{f}_j^{\mathsf{T}} \mathbf{x}_{(c)}, z_{jc}). \end{aligned} \tag{2}$$

There are two choices for the form of ϕ_j: the multiplicative form $\phi_j(u, z) = \exp\left(-\frac{1}{2} u^2 z - \psi_m(z)\right)$ [11] and the additive form $\phi_j(u, z) = \exp\left(-b(u - z)^2 - \psi_a(z)\right)$ [12]. In either case, $\psi_m(z)$ respectively $\psi_a(z)$ are chosen such that $\rho_j(u) = \bigoplus_z \phi_j(u, z)$ (see example in Fig. 1). The name stems from the fact that the energy of $\phi_j(u, z)$ is quadratic in u but not in z. This further implies that for a fixed setting of $\mathbf{z}$ the distribution $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{z}})$ is jointly
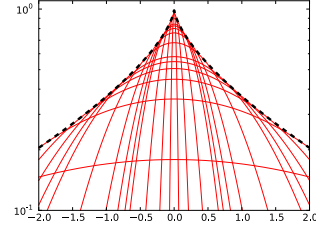

Figure 1. **Half-quadratic representation of a sparse potential.** Hyper-Laplacian $\rho(u) = \exp(-|u|^\gamma)$ (dashed, black), where $\rho(u) = \sup_z \phi(u, z)$, and $\phi(u, z) = \exp(-\frac{1}{2} u^2 z - \psi_m(z))$ (solid, red) with $\psi_m(z) = (1 - \gamma/2) \cdot (z/\gamma)^{\frac{\gamma}{\gamma-2}}$ and $\gamma = 2/3$.

Gaussian. When combined with a Gaussian likelihood, we obtain a Gaussian posterior for a fixed setting of $\mathbf{z}$:

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}, \mathbf{z}) &\propto \mathcal{N}(\mathbf{y}; \mathbf{Kx}, \sigma^2 \mathbf{I}) \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{z}}) \\ &\propto \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}, \mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}, \mathbf{z}}). \end{aligned} \tag{3}$$

The benefit of this is that MAP estimation can now be carried out on the augmented posterior $p(\mathbf{x}, \mathbf{z}|\mathbf{y})$ with a variational EM algorithm [*cf.* 22] that alternates between maximizing $p(\mathbf{x}|\mathbf{y}, \mathbf{z})$ and using $p(\mathbf{z}|\mathbf{x}, \mathbf{y})$ to update the auxiliary variables based on the operation $\bigoplus$. Maximizing $p(\mathbf{x}|\mathbf{y}, \mathbf{z})$ w.r.t. $\mathbf{x}$ amounts to computing $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}, \mathbf{z}}$, which requires solving a sparse linear equation system. Updating $\mathbf{z}$ based on $p(\mathbf{z}|\mathbf{y}, \mathbf{x})$ is easy, since all z_{jc} are independent:

$$p(\mathbf{z}|\mathbf{x}, \mathbf{y}) \propto \prod_j \prod_{c \in \mathcal{C}_j} p(z_{jc}|\mathbf{x}, \mathbf{y}) \tag{4}$$

$$p(z_{jc}|\mathbf{x}, \mathbf{y}) \propto \phi_j(\mathbf{f}_j^{\mathsf{T}} \mathbf{x}_{(c)}, z_{jc}). \tag{5}$$

By using the fact that a wide variety of robust (sparse) potentials ρ_j can be expressed (or approximated) by taking the supremum over the auxiliary variables $\mathbf{z}$ [3], one can reformulate the majority of sparse image priors in this way. Levin *et al.* [18] and Krishnan and Fergus [17] have employed this principle for efficient image deblurring. Note that MRF image priors based on Gaussian scale mixtures (GSMs) [29] can also be interpreted as an instance of half-quadratic regularization in which $\bigoplus = \sum$ (or $\bigoplus = \int$ for infinite GSMs). This has been used by Schmidt *et al.* [25] for deblurring with sampling-based inference, which alternates between sampling from $p(\mathbf{x}|\mathbf{y}, \mathbf{z})$ and $p(\mathbf{z}|\mathbf{x}, \mathbf{y})$.

2.1. Discriminative alternative

To see how classical half-quadratic regularization can be connected to a discriminative approach, it is instructive to consider what happens during the last inference iteration. Once the final set of latent variables $\mathbf{z}^*$ has been determined from Eq. (5), the output image $\mathbf{x}^*$ is inferred by maximizing $p(\mathbf{x}|\mathbf{y}, \mathbf{z}^*)$ from Eq. (3). This distribution is nothing but an anisotropic (or inhomogeneous) Gaussian random field, whose mean and covariance depend on $\mathbf{y}$ and $\mathbf{z}^*$.

---

[1]To improve readability, related work will be reviewed in-line.

Therefore $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y},\mathbf{z}^*}$ and $\boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y},\mathbf{z}^*}$ are the mean and covariance parameters of a multivariate normal distribution defined on the whole image, chosen through $\mathbf{z}^*$ so as to hopefully lead to good deblurring results. The value of $\mathbf{z}^*$ depends on the specific choice of potential functions ρ_j and their half-quadratic inference approximations ϕ_j (Eq. (5)).

It is now natural to ask whether we can instead directly regress the Gaussian random field parameters from the input image. To this end we can regress a precision matrix $\boldsymbol{\Theta}(\mathbf{y})$ and a vector $\boldsymbol{\theta}(\mathbf{y})$, and set $\boldsymbol{\mu} := [\boldsymbol{\Theta}(\mathbf{y})]^{-1}\boldsymbol{\theta}$ and $\boldsymbol{\Sigma} := [\boldsymbol{\Theta}(\mathbf{y})]^{-1}$. Then the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$ are learned functions of the observed image $\mathbf{y}$. There are three potential advantages: *First*, we avoid the expensive iterative computation of the half-quadratic optimization. *Second*, we can regress the parameters discriminatively in order to optimize a given performance measure, such as PSNR. *Third*, we are no longer constrained to the form of Eq. (5) so that we can now use an expressive regression model on the input image. That is, we are not restricting[2] the resulting model compared to Eq. (3); in fact, we can potentially learn a more expressive model.

We arrived at this model from a novel analysis of the half-quadratic approximation, but predicting the means and covariances of a Gaussian model has been done before: Gaussian conditional random fields, first proposed by Tappen *et al.* [27], have led to competitive results in image denoising. A recent extension we will build on are the regression tree fields (RTF) by Jancsary *et al.* [14, 15].

Image deblurring. Non-blind image deblurring is more difficult than image denoising, and it might be difficult to directly regress suitable model parameters. To illustrate this difficulty, let us assume that $\mathbf{f}_j$ are first-order derivative filters. Then, in the generative approach one can think of z_{jc} as modulating pairwise potentials: reducing smoothness constraints in case of large image derivatives *of the output image* $\mathbf{x}$, and imposing smoothness otherwise. In other words, in the generative approach $\mathbf{z}$ determines the local model of the restored image $\mathbf{x}$. Both $\mathbf{x}$ and $\mathbf{z}$ are iteratively refined via half-quadratic inference. In a discriminative model we have access only to the corrupted image $\mathbf{y}$ in order to determine suitable local models.

But in the case of deblurring, the image content in $\mathbf{y}$ is shifted and combined with other parts of the image, depending on a blur kernel that is different for each image. This makes the choice of local models difficult. We believe this is one of the reasons why discriminative non-blind deblurring approaches had not been attempted before.

Figure 2. **Standard half-quadratic vs. discriminative cascade.** In a standard half-quadratic approach *(top)*, each z_{jc} can only be updated via Eq. (5) based on the filter response $\mathbf{f}_j^{\mathrm{T}}\mathbf{x}_{(c)}$ of the pixels in the local clique (small white circles, only one filtered image $\mathbf{f}_j \otimes \mathbf{x}$ shown). In the proposed discriminative cascade *(bottom)*, one can use arbitrary features of the image over larger areas (large white circles) to find model parameters $\boldsymbol{\Theta}^{(i)}$ and $\boldsymbol{\theta}^{(i)}$ via regression. At each stage, the functions $\boldsymbol{\Theta}^{(i)}$ and $\boldsymbol{\theta}^{(i)}$ depend on $\mathbf{y}$ through features, such as filter bank responses, image intensities, and also depend on all $\mathbf{x}^{(i)}$ from previous iterations (not shown).

2.2. Discriminative model cascade

To build a discriminative model for deblurring, we draw inspiration from the iterative refinement of $\mathbf{z}$ in half-quadratic regularization. We start with an educated guess of the Gaussian model parameters, regressed from the input image, to obtain a restored image $\mathbf{x}^{(1)}$, which is less corrupted than the original input image. We can then use this as an intermediate result to help regress refined Gaussian model parameters, in order to obtain a better restored image $\mathbf{x}^{(2)}$, *etc.*, effectively obtaining a cascade of refined models. Note that this is a general approach that is not only applicable to image deblurring. Image denoising and other restoration tasks may also benefit from such a model cascade and repeated refinement of the auxiliary variables; we do not consider this here, however.

A key advantage of a discriminative approach for predicting model parameters $\boldsymbol{\Theta}^{(i)}$, $\boldsymbol{\theta}^{(i)}$ at the ith stage is its flexibility. As discussed above, a standard generative half-quadratic approach updates each z_{jc} only based on the local clique of the current estimate of the restored image (*cf.* Eq. (5)). In a discriminative approach, we can regress the parameters based on arbitrary local and global properties of the input image as well as the current estimate of the re-

---

[2] Note that since a multivariate Gaussian distribution can always be expressed as a product of bivariate Gaussians, the final MAP estimate in half-quadratic regularization comes from a pairwise MRF *even if the corresponding sparse image prior models high-order interactions*. This does not mean, however, that high-order dependencies are ignored. They are only hidden in the estimate $\mathbf{z}^*$.
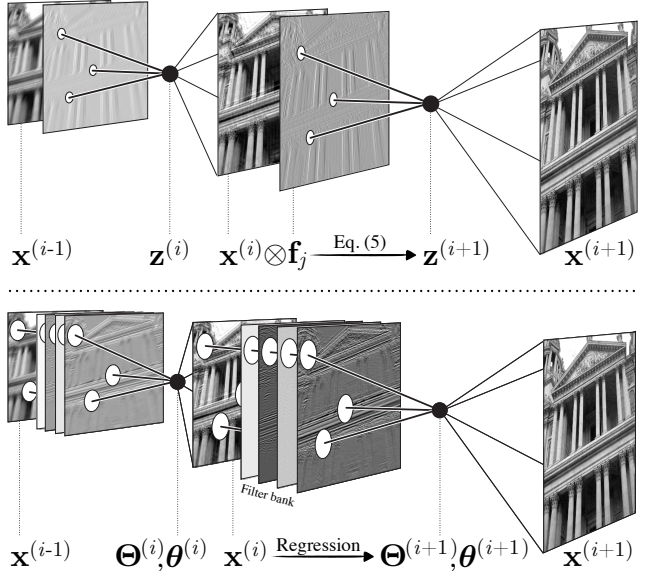
stored image (see Fig. 2 for an illustration). Consequently, we can expect to obtain better estimates in fewer iterations, compared to a non-discriminative approach.

Other related work. This iterative procedure can also be linked with earlier ideas about iterative refinement. The idea of *auto-context* [28] is to use the same probabilistic model multiple times in sequence, where each model receives as input the observed image and the output of the previous model in the sequence. The proposed discriminative cascade is also related to the active random field [2], which is a multi-stage approach for image denoising that is trained discriminatively. The key difference is that each stage in [2] corresponds to a gradient descent iteration of the model energy, whereas we use full discriminative prediction.

3. Gaussian CRF for Deblurring

As we have seen, a discriminative alternative to half-quadratic MAP estimation is conceptually attractive, but also challenging due to the need of regressing local image models from the blurred input image $\mathbf{y}$. To address this challenge we propose a novel Gaussian CRF $p(\mathbf{x}|\mathbf{y}, \mathbf{K})$ for non-blind image deblurring. One challenge in devising such a model is that we cannot train a different model for every blur matrix $\mathbf{K}$; this difficulty may in fact be the reason why no such approach exists to date. To see how this can be circumvented, we can take inspiration from generative approaches to deblurring and see how the Gaussian blur likelihood $p(\mathbf{y}|\mathbf{x}, \mathbf{K})$ contributes to the posterior distribution when assuming a Gaussian prior:

$$
\begin{aligned}
p(\mathbf{x}|\mathbf{y}, \mathbf{K}) &\propto p(\mathbf{y}|\mathbf{x}, \mathbf{K}) \cdot p(\mathbf{x}) \\
&\propto \mathcal{N}(\mathbf{y}; \mathbf{Kx}, \mathbf{I}/\alpha) \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\Theta}^{-1}\boldsymbol{\theta}, \boldsymbol{\Theta}^{-1}) \\
&\propto \mathcal{N}\big(\mathbf{x}; (\alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}\alpha\mathbf{K}^{\mathrm{T}}\mathbf{y}, (\alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}\big) \\
&\quad \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\Theta}^{-1}\boldsymbol{\theta}, \boldsymbol{\Theta}^{-1}) \\
&\propto \mathcal{N}\big(\mathbf{x}; (\boldsymbol{\Theta} + \alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}(\boldsymbol{\theta} + \alpha\mathbf{K}^{\mathrm{T}}\mathbf{y}), \quad (6) \\
&\quad (\boldsymbol{\Theta} + \alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}\big),
\end{aligned}
$$

where $\alpha = 1/\sigma^2$ relates to the noise level, $\boldsymbol{\Theta}$ is the precision of the Gaussian prior, and $\boldsymbol{\theta}$ relates to its mean. We can now define a Gaussian CRF in which the model parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\theta}$ are not fixed, but regressed from the input image, *i.e.* $\boldsymbol{\Theta} \equiv \boldsymbol{\Theta}(\mathbf{y})$ and $\boldsymbol{\theta} \equiv \boldsymbol{\theta}(\mathbf{y})$ are functions of $\mathbf{y}$. Note that the CRF is parameterized by a fixed blur $\mathbf{K}$ as in Eq. (6); the blur is *not* used as an input feature to the regressor.

Please note that for the problem of image denoising, *i.e.* $\mathbf{K} = \mathbf{I}$, explicitly incorporating a component related to the likelihood as described above is not necessary, since its contribution can be learned by the regression function. However, for deblurring this is not feasible, and it is crucial to incorporate a blur component into the model to adapt to arbitrary blurs.

Once we have determined the parameters via regression, we can obtain a deblurred image as the MAP estimate, which can be derived in closed form as the mean of the Gaussian CRF,

$$
\begin{aligned}
\arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}, \mathbf{K}) &= \\
(\boldsymbol{\Theta}(\mathbf{y}) + \alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}&(\boldsymbol{\theta}(\mathbf{y}) + \alpha\mathbf{K}^{\mathrm{T}}\mathbf{y}), \quad (7)
\end{aligned}
$$

and can be computed by solving a sparse linear system.

3.1. Regression tree field

To make our approach concrete, we need to specify the regression functions $\boldsymbol{\Theta}(\mathbf{y})$ and $\boldsymbol{\theta}(\mathbf{y})$. To that end, we draw on the recently proposed regression tree field (RTF) model by Jancsary *et al.* [14, 15]. RTFs have shown excellent results in image restoration applications, such as image denoising, inpainting, and colorization.

In general, RTFs take the form of a Gaussian CRF in which a non-linear regressor is used to specify the local model parameters. Specifically, regression trees are used, where each leaf stores an individual linear regressor that determines a local potential. Since any Gaussian CRF can be decomposed into factors relating no more than two pixels, our posterior density attains the following form:

$$
p(\mathbf{x}|\mathbf{y}, \mathbf{K}) \propto \mathcal{N}(\mathbf{y}; \mathbf{Kx}, \mathbf{I}/\alpha) \cdot \prod_{j=1}^{J+1} \prod_{c \in \mathcal{C}_j} \phi_j(\mathbf{x}_{(c)}, \mathbf{y}) \quad (8)
$$

$$
\phi_j(\mathbf{x}_{(c)}, \mathbf{y}) \propto \exp\left(-\frac{1}{2}\mathbf{x}_{(c)}^{\mathrm{T}}\boldsymbol{\Theta}_c^j(\mathbf{y})\mathbf{x}_{(c)} + \mathbf{x}_{(c)}^{\mathrm{T}}\boldsymbol{\theta}_c^j(\mathbf{y})\right),
$$

where $\mathcal{C}_j$ denotes all pairs of neighboring pixels in the jth of J possible spatial configurations. Concretely, we use 8- and 24-neighborhoods depending on the stage in our prediction cascade, *i.e.* $J = 4$ and $J = 12$, respectively (due to spatial symmetries). Additionally, at each stage, we employ a single unary potential $\phi_{J+1}(\mathbf{x}_{(c)}, \mathbf{y})$, where $\mathcal{C}_{J+1}$ is simply the set of all individual pixels.

We extend these previous RTF-based approaches to our setting by *(a)* incorporating the blur likelihood for non-blind image deblurring into the prediction as outlined in Eqs. (6) and (7), and *(b)* by assembling multiple RTFs into a model cascade that iteratively refines the prediction (see Sec. 3.3).

Note that the RTF generalizes the Gaussian CRF of Tappen *et al.* [27] in two ways: First, the potentials of an RTF are non-linearly dependent on the input image via non-parametric regression trees. Second, the model parameters of arbitrary pairwise Gaussian potentials (with full mean and covariance) are regressed from the input image, whereas [27] restrict their parameterization to diagonal weighting of filter responses.

3.2. Training

While probabilistic training is possible [15], we here follow [14] and learn the regressors using loss-based train-
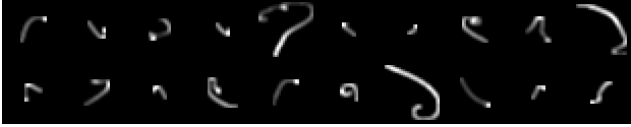
Figure 3. **Examples of artificially generated blur kernels.**

Figure 4. **RTF prediction cascade.** Only three stages are shown.

ing, in particular, such that the peak signal-to-noise ratio (PSNR) is maximized. All parameters of the model, including the split functions in the tree and the linear regressors in the leaves, are chosen to maximize PSNR.

Discriminative training necessitates a sufficient amount of training data to ensure generalization. Since capturing image pairs of blurred and clean images is difficult, one possible avenue is to synthesize training data by blurring clean images with realistic blurs. Unfortunately, existing databases [16, 20] only supply a relatively limited number of blur kernels, and moreover serve also for testing. Hence the model should not be trained on these. We address this problem by generating realistic-looking blur kernels by sampling random 3D trajectories using a simple linear motion model; the obtained trajectories are projected and rasterized to random square kernel sizes in the range from 5×5 up to 27×27 pixels (see Fig. 3). While it would of course be possible to create even more realistic kernels through more accurate models of camera shake motion[3], we find that these synthetic kernels already allow to generalize well to unseen real blur (*cf.* Sec. 4). We synthetically generate blurred images by convolving each clean image with an artificially generated blur kernel, and subsequently add pixel-independent Gaussian noise (using standard deviations $\sigma = 2.55$ or 0.5, see experiments in Sec. 4). We use crops of 128×128 pixels from the training portion of the Berkeley segmentation dataset [1] as ground truth images.

3.3. RTF prediction cascade

As argued in Sec. 2, it is difficult to directly regress good local image models from the blurred input image. Therefore, we employ a cascade of RTF models, where each subsequent model stage uses the output of all previous models as features for the regression (see Fig. 4 for an illustration).

We train the first stage of the cascade with minimal conditioning on the input image to avoid overfitting. The parameters of the unary and pairwise potentials are only linearly regressed from the pixels in the respective cliques (plus a constant pseudo-input); we do not use a regression tree. We further use an 8-connected graph structure, resulting in one unary and four pairwise potentials (horizontal, vertical, and two diagonals). We train this model with 200 pairs of blurred and clean images, which is ample since there are only few model parameters. This model will be referred to as RTF_1.
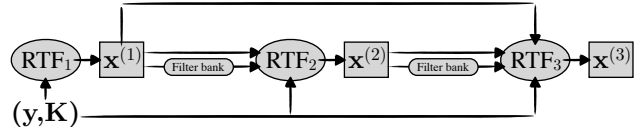
While we do not expect excellent results from RTF_1, it is able to remove the dominant blur from the input image (*cf.* Sec. 4 and Fig. 6) and makes it much easier for subsequent RTF stages to regress good potentials for the CRF. Besides the blurred input image, the second stage, RTF_2, thus uses the output of RTF_1 as input feature. Additionally, we evaluate a filter bank on the output of RTF_1 to obtain more expressive features. In doing so we follow [14], which obtained improved results in image denoising using the output of a filter bank as input to the regressor. However, we use a different filter bank here, the 16 generatively trained 5×5 filters from the recent Fields-of-Experts model of [10]; we found these filters to outperform other filter banks we have tried, including those used in [14][4].

We use all these features for the split tests in the regression tree (non-linear regression), as well as for the linear potential parameter regressor that is stored in each leaf of the tree. We choose regression trees of depth 7. All subsequent model stages, *i.e.* RTF_3, RTF_4, *etc.*, take as features the outputs from all previous stages, where the filter bank is always only evaluated on the directly preceding model output. Please see Fig. 4 for a schematic. Starting with RTF_2, the Gaussian CRF at each layer uses a 24-connected graph structure, *i.e.* each pixel is connected to all others in a 5×5 neighborhood. Due to the increased amount of model parameters, we train RTF_2 and each subsequent stage with 500 training images, randomly cropped from the training portion of the Berkeley segmentation dataset [1] and blurred with randomly chosen artificial blur kernels (different at each stage).

An interesting property of our model cascade is that it yields a deblurred image after every stage, not only at the end. Even if a deep cascade was trained, at test time we can trade off computational resources versus quality of the deblurred image by stopping after a certain stage (*cf.* Fig. 6; see [9] for a similar approach applied to segmentation).

4. Experiments

To demonstrate the performance of our approach, we apply it to three challenging datasets, specifically to highlight individual benefits. First, we analyze the performance in standard conditions for non-blind deblurring, *i.e.* when training and testing is carried out with perfect blur kernels. Second, we evaluate the generalization ability of our ap-

---

[3]We think that on average these synthetic blur kernels may in fact be more challenging than typical real ones.

[4]There may be an interesting relation to generative pre-training for deep belief networks [13]; we leave this for future work.

Method	σ		Stage	σ	
	2.55	7.65		2.55	7.65
Lucy-Richardson [21, 23]	25.38	21.85	RTF_1	26.33	24.23
Krishnan and Fergus [17]	26.97	24.91	RTF_2	28.21	25.54
Levin *et al.* [18]	28.03	25.36	RTF_3	28.50	25.75
5×5 FoE (MAP) [24]	28.44	25.66	RTF_4	28.58	25.81
Pairw. MRF (MMSE) [25]	28.24	25.63	RTF_5	28.65	25.87
3×3 FoE (MMSE) [25]	28.66	25.68	RTF_6	**28.67**	**25.89**

Table 1. Average PSNR (dB) on 64 images from [25] for two noise levels. Left half reproduced from [25] for ease of comparison.

proach by training the model to deal with imperfect blur kernels. This is important for blind deblurring, where the estimated blur kernels mostly contain some errors. Third, we demonstrate the applicability to realistic higher-resolution images. Please note that images and kernels are always kept strictly separate for training and testing in all experiments.

Standard conditions. We trained a six-stage RTF prediction cascade as described in Sec. 3 and evaluate all stages individually on 64 test images taken from [25]. Training images have been blurred synthetically with 1% additive white Gaussian noise ($\sigma = 2.55$); test images with both $\sigma = 2.55$ and a higher noise level of $\sigma = 7.65$. While we used artifical blur kernels to generate our training data, the test images from [25] have been created with the realistic kernels from [20]. The blur kernels used for deblurring are slightly perturbed from the ground truth to mimic kernel estimation errors, but the perturbation is somewhat minor here. We compare our average PSNR performance to all methods that were evaluated in [25]. The results in Tab. 1 show that we achieve state-of-the-art performance that is on par with the high-quality sampling-based approach of Schmidt *et al.* [25] at $\sigma = 2.55$, and even outperforms it at $\sigma = 7.65$ despite not being trained for this noise level (only α was adapted, see Eq. (6)). As we shall discuss below, our approach is many times faster, however. At the noise level our model is trained for (2.55), we strongly outperform the efficient half-quadratic regularization approach of Krishnan and Fergus [17] by over 1.5dB, and the popular method of Levin *et al.* [18] by 0.6dB. The clear performance gains at the higher noise level demonstrate our model's noise generalization. We further notice that the weakly conditional first stage (RTF_1) leads only to modest performance levels here; RTF_2 and RTF_3 boost the performance substantially further. Later stages lead to additional gains, but less so. Aside from the raw numbers, it is noteworthy that our model is able to preserve small details, while at the same time reconstructing smooth areas well (see Fig. 5 for an example). Note that this is not the case for the approaches tested in [25].

This demonstrates that when testing (and training) is done with the correct (*i.e.* ground truth) blur kernels, as usual for non-blind deblurring, our approach achieves excellent results. Another important point is that even though

(a) [25], PSNR = 29.05dB   (b) RTF_6, PSNR = 29.23dB

Figure 5. **Deblurring example (cropped).** Qualitative comparison with the high-quality approach of [25] (3×3 FoE, MMSE estimation). Our approach *(b)* reconstructs smooth and textured areas well, exhibits fewer artifacts, and is many times faster. *Best viewed zoomed in on screen.*

our model has been trained on artificially generated blur kernels (Fig. 3), it apparently generalizes well to real blurs.

Adaptation to kernel estimation errors. Blind deblurring approaches often produce kernel estimates with substantial errors, which can cause ringing artifacts in the restored image [*cf*. 31]. Hence, it is important to evaluate and adapt our model to this realistic scenario. To train our model for this task, we experimented with adding noise to the ground truth kernels and also used estimated kernels for training.

We consider the data of Levin *et al.* [19] as a benchmark, which provides several kernel estimates besides blurred and ground truth images for 32 test instances, as well as deblurring results with the various kernel estimates. Since the amount of noise in these blurred images is significantly lower than in the benchmark of [25], we only added Gaussian noise with $\sigma = 0.5$ to our training images. We evaluate average PSNR performance over all 32 images (using code by [19] to account for translations in kernel estimates) instead of error ratios as in [19], since we are not interested in the quality of the estimated kernels itself, but rather the final restoration performance given the estimated kernels.

The results in Tab. 2 show that training with ground truth kernels leads to subpar performance when kernels estimates are used at test time. Adding noise to the ground truth kernels for training leads to improved results of RTF_1 with estimated kernels at test time, but performance of our second stage model RTF_2 already deteriorates; hence those noisy kernels are not an ideal proxy for real kernel estimates. However, we achieve superior results by training our model with a mix of perfect and estimated kernels (obtained with the method of Xu and Jia [30]), *i.e.* for half of the synthetically blurred training images we use an estimated kernel instead of the ground truth kernel[5]. Compared to the deblurred images from [19] (which used the

---

[5]Here, we trained RTF_1 and RTF_2 with the same 200 images as it was time-consuming to obtain good enough kernel estimates for training.

Method	Kernels for training	Kernels for testing					
		GT	Levin *et al.* [19]	Cho and Lee [6]	Fergus *et al.* [8]	Noisy GT	Xu and Jia [30]
RTF$_1$	GT	32.76	29.41	28.29	27.86	26.67	29.04
RTF$_2$	GT	33.81	29.52	27.76	27.84	26.52	28.29
RTF$_1$	Noisy GT	32.08	29.73	29.36	28.49	28.69	30.25
RTF$_2$	Noisy GT	30.51	29.03	28.75	27.58	30.34	29.56
RTF$_1$	Mix of GT & [30]	32.90	29.90	29.33	28.63	28.10	30.30
RTF$_2$	Mix of GT & [30]	33.97	30.40	29.73	29.10	28.07	30.84
[18]	—	32.73	30.05	29.71[6]	28.38	—	—

Table 2. Average results (PSNR in dB) on 32 images from [19]. All kernels used for testing are from the benchmark set [19], except for those in the two rightmost columns: we derived the noisy ground truth (GT) kernels from the provided GT kernels, and estimated kernels with [30]. The last row shows the average performance of deblurring results provided by [19] (using the non-blind approach of [18]). For the kernel estimates of [19] (4th column), we used the "free energy with diagonal covariance approximation" algorithm in the filter domain.

non-blind approach of [18]), we achieve substantial performance improvements for deblurring with estimated kernels of up to 0.72dB (for kernels from [8]). Furthermore, it is interesting to note that the first stage of our model already achieves good performance; this is presumably due to the much reduced amount of noise in this benchmark[7].

Runtime. The computational demand of our method is comparable to the half-quadratic approach of [18], but uses this computational budget more effectively due to its discriminative nature (*cf*. Sec. 2 and Fig. 2). Also note that the tree-based regressor is very efficient. As a result, we achieve state-of-the-art performance on par with the best result of [25], but much faster: about 2 seconds per image in Tab. 1 (all six model stages combined) compared to 4 minutes for [25]. For the benchmark in Tab. 2 with larger images, we require around 3 seconds for each model stage.

Realistic higher-resolution images. We consider the recent benchmark for camera shake by Köhler *et al.* [16] to demonstrate results on realistic higher-resolution images; these images may substantially violate our model's stationary blur and Gaussian noise assumptions (which can deteriorate performance [*cf*. 7, 26]). The benchmark consists of 4 color images of size 800×800 pixels blurred by 12 different real camera motions, yielding 48 images in total. The overall best performing blind deblurring approach in this benchmark is the one by Xu and Jia [30] despite making a stationary blur assumption, *i.e.* the same blur kernel is used in all parts of the image. We use the provided kernel estimates by [30] from the benchmark dataset, but with our non-blind method to obtain the deblurred image (treating color channels R, G, and B independently). Tab. 3 shows that performance (evaluated using the provided code) can substantially be improved by using our RTF$_2$ model instead of their non-blind step (which is related to [17]). While Xu

	Image 1	Image 2	Image 3	Image 4
Kernel 1	+0.44	+0.54	+1.05	+0.76
Kernel 2	+0.44	+0.27	+0.38	+0.46
Kernel 3	+0.02	+0.03	+0.39	−0.26
Kernel 4	+0.31	+0.30	+0.61	+0.27
Kernel 5	+0.61	+0.44	+0.64	+0.05
Kernel 6	+0.40	+0.41	+1.03	+0.48
Kernel 7	+0.24	+0.55	+0.45	+0.31
Kernel 8	+0.76	+0.56	+2.17	+1.73
Kernel 9	+0.35	−0.09	+0.02	+0.23
Kernel 10	+0.19	−0.55	+0.25	+0.29
Kernel 11	−0.19	−0.43	+0.46	+0.09
Kernel 12	+0.76	+0.04	+0.66	+0.64

Table 3. Performance gain (PSNR in dB) over the results of Xu and Jia [30] in the benchmark of Köhler *et al.* [16] for each combination of 4 test images and 12 blur kernels. We use the provided blur kernel estimates of [30] with our RTF$_2$ model for non-blind deblurring. We can improve the performance in 43 of 48 test instances, on average about 0.41dB.

Figure 7. Deblurring example from the benchmark of [16] (*cf*. Tab. 3), showing the result of our RTF$_2$ model *(right)* given the blurred image *(left)* and the kernel estimates by [30].

and Jia's non-blind step is inherently faster, it does lead to substantially worse results, here on average 0.41dB. Fig. 7 shows an example of a deblurred image. Note that the RTF$_2$ model used here is the same as in Tab. 2, *i.e.* trained with a mix of ground truth and estimated kernels (using [30]), and additive Gaussian noise with $\sigma = 0.5$.

---

[6]The result might not be fully comparable, since the blur kernel estimation *and* non-blind method from [6] may have been used.

[7]Theoretically, in the absence of noise non-blind deblurring can be solved exactly without any regularization by inverting the blur matrix.
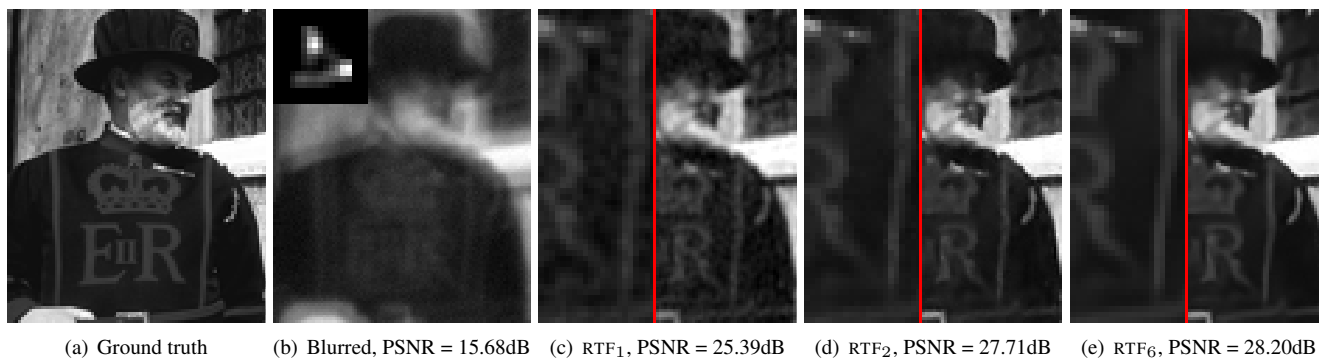
| (a) Ground truth | (b) Blurred, PSNR = 15.68dB | (c) RTF$_1$, PSNR = 25.39dB | (d) RTF$_2$, PSNR = 27.71dB | (e) RTF$_6$, PSNR = 28.20dB |

Figure 6. **Deblurring example at different model stages.** The first stage RTF$_1$ removes dominant blur from the image *(c)*, but artifacts remain. The second stage RTF$_2$ *(d)* substantially improves upon this result quantitatively and qualitatively. Further model stages continue to suppress noise and refine image details *(e)*. The left sides of *(c–e)* show a closeup view of image details on the respective right sides. The blur kernel is shown at the upper left of *(b)*, scaled and resized for better visualization. *Best viewed on screen.*

5. Summary and Conclusions

From a novel analysis of common half-quadratic regularization, we introduced – to the best of our knowledge – the first discriminative non-blind deblurring method. Our proposed cascade model is based on regression tree fields at each stage, which are trained by loss minimization on training data generated with synthesized blur kernels. We demonstrated state-of-the art performance on three challenging benchmarks, including robustness to kernel estimation errors in the context of blind deblurring. Our approach is not limited to image deblurring and can readily be extended to other image restoration applications in the future.

References

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

[2] A. Barbu. Learning real-time MRF inference for image denoising. *CVPR 2009*.

[3] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.

[4] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? *CVPR 2012*.

[5] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *ICIP 1994*.

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

[7] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. *ICCV 2011*.

[8] 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), 2006.

[9] B. Fröhlich, E. Rodner, and J. Denzler. As time goes by—anytime semantic segmentation with iterative context forests. *Pattern Recognition (DAGM) 2012*.

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

[11] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *TPAMI*, 14(3):367–383, 1992.

[12] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE TIP*, 4(7):932–946, 1995.

[13] G. A. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18:1527–1554, 2006.

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

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

[16] 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. *ECCV 2012*.

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

[18] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM T. Graphics*, 26(3):70:1–70:9, 2007.

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

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

[21] L. B. Lucy. An iterative technique for the rectification of observed distributions. *The Astronom. Journal*, 79:745, 1974.

[22] J. A. Palmer, D. P. Wipf, K. Kreutz-Delgado, and B. D. Rao. Variational EM algorithms for non-Gaussian latent variable models. *NIPS*2005*.

[23] W. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. America*, 62(1):55–59, 1972.

[24] S. Roth and M. J. Black. Fields of experts. *IJCV*, 82(2):205–229, 2009.

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

[26] Y.-W. Tai and S. Lin. Motion-aware noise filtering for deblurring of noisy and blurry images. *CVPR 2012*.

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

[28] Z. Tu. Auto-context and its application to high-level vision tasks. *CVPR 2008*.

[29] M. J. Wainwright and E. P. Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. *NIPS*1999*.

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

[31] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM T. Graphics*, 27(3), 2008.