Discriminative Non-blind Deblurring

Uwe Schmidt[1] Carsten Rother[2] Sebastian Nowozin[2] Jeremy Jancsary[2] Stefan Roth[1]

[1] Department of Computer Science, TU Darmstadt [2] Microsoft Research Cambridge

Motivation

- *Non-blind deblurring* is an important component for removing image blur (*e.g.* due to camera shake) after blur estimation.
- High-quality learning-based methods have been limited to the generative case and are often computationally expensive.
- Hand-defined models with inferior quality are most widely used.
- How to devise a flexible discriminative approach with high restoration quality and efficiency?

Three challenges:

1. Lack of training data, in particular realistic blur kernels
2. Work with arbitrary images and blurs
3. Appropriate feature functions given blurred image

1. Synthesize Training Data

- Realistic blur kernels are scarce, recording them is difficult
- Existing blurs used for testing, shouldn't be used for training
- **Generate artificial blur kernels** from random 3D trajectories, obtained with simple motion model
- Blurred image synthesized from clean image and blur kernel

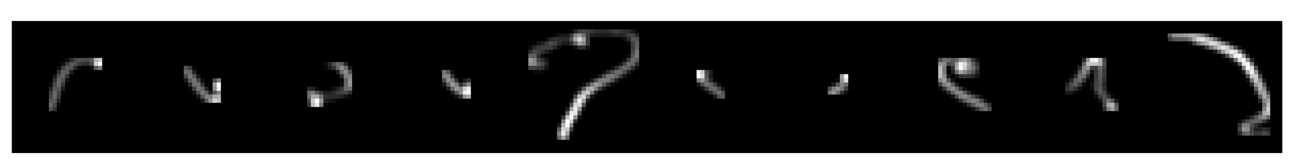


Figure 1: *Examples of artificially generated blur kernels.*

2. Gaussian CRF for Deblurring

Idea: **Split parameters into learnable and blur-dependent ones**, akin to combining likelihood and prior in a generative approach:

$$p(\mathbf{x}|\mathbf{y},\mathbf{K}) \propto \underbrace{\mathcal{N}(\mathbf{y};\mathbf{K}\mathbf{x},\mathbf{I}/\alpha)}_{\text{Likelihood}} \cdot \underbrace{\mathcal{N}(\mathbf{x};\boldsymbol{\Theta}^{-1}\boldsymbol{\theta},\boldsymbol{\Theta}^{-1})}_{\text{Prior}}$$

$$\propto \mathcal{N}(\mathbf{x};(\boldsymbol{\Theta}+\alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}(\boldsymbol{\theta}+\alpha\mathbf{K}^{\mathrm{T}}\mathbf{y}),(\boldsymbol{\Theta}+\alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1})$$

Now define Gaussian CRF where model parameters are regressed from blurred input image $\mathbf{y}$, *i.e.* $\boldsymbol{\Theta} \equiv \boldsymbol{\Theta}(\mathbf{y})$ and $\boldsymbol{\theta} \equiv \boldsymbol{\theta}(\mathbf{y})$. The CRF is parametrized by and thus works with arbitrary blurs $\mathbf{K}$ and images $\mathbf{y}$.

Deblurred image $\hat{\mathbf{x}}$ obtained as MAP estimate of Gaussian CRF:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y},\mathbf{K}) = (\boldsymbol{\Theta}(\mathbf{y})+\alpha\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}(\boldsymbol{\theta}(\mathbf{y})+\alpha\mathbf{K}^{\mathrm{T}}\mathbf{y})$$

Use **regression tree fields** (RTFs) [3, 4] to learn model parameters $\boldsymbol{\Theta}(\mathbf{y})$ and $\boldsymbol{\theta}(\mathbf{y})$.

- RTFs are flexible Gaussian CRFs
- Non-linear regression via regression trees
- Loss-based training [3] (for PSNR)

Extend previous RTFs by *1)* incorporating blur parameters, and *2)* using a cascade.

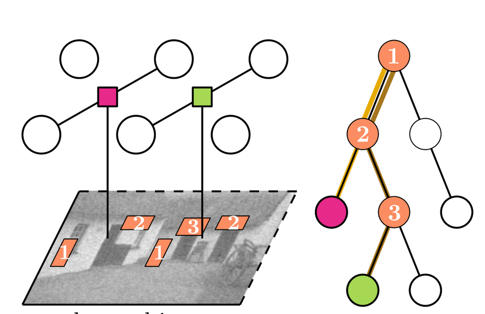


Figure 2: *RTF example* from [3].

Our Approach

- **First discriminative approach for non-blind deblurring** for arbitrary images and blurs.
- **Efficient** with **state-of-the-art results** on three benchmarks.
- **Generalizes** commonly-used **half-quadratic deblurring**.
- **Cascade model** with a Gaussian CRF at each stage, based on recent *regression tree fields* (RTFs).
- Loss-based training with data from *synthetic blur kernels*.
- Cascade model **not limited to image deblurring**.

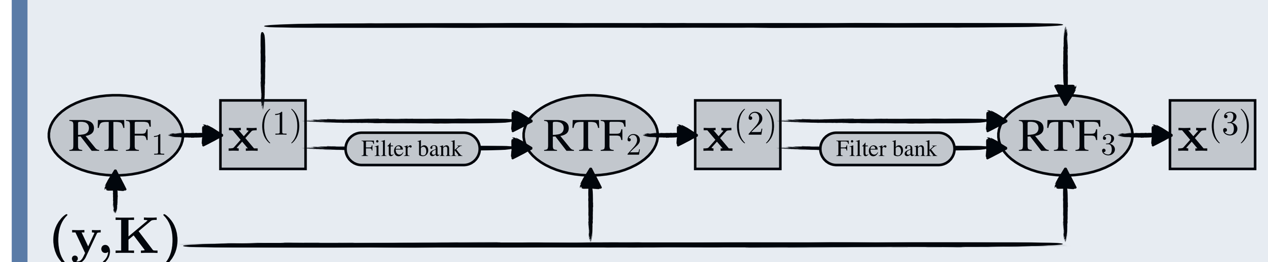


Figure 3: **RTF prediction cascade.** Only three stages are shown.

3. Discriminative Prediction Cascade

Difficult to devise feature functions due to blurred image content $\mathbf{y}$. Easier for denoising, where discriminative methods successful, *e.g.* [3, 11].

Motivation: **Half-quadratic deblurring** as commonly-used to ease inference with sparse image priors $p(\mathbf{x})$ [6, 7, 10]:

- Latent variables $\mathbf{z}$ introduced with $p(\mathbf{x}) = \max_{\mathbf{z}} p(\mathbf{x},\mathbf{z})$; augmented posterior:
$$p(\mathbf{x},\mathbf{z}|\mathbf{y},\mathbf{K}) \propto p(\mathbf{y}|\mathbf{x},\mathbf{K}) \cdot p(\mathbf{x},\mathbf{z})$$
- MAP estimation of deblurred image $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y},\mathbf{K})$ via alternating max. of $p(\mathbf{x}|\mathbf{y},\mathbf{z},\mathbf{K})$ and $p(\mathbf{z}|\mathbf{x},\mathbf{y},\mathbf{K})$.
- Iterative refinement of *inhomogeneous Gaussian MRF* $p(\mathbf{x}|\mathbf{y},\mathbf{z},\mathbf{K})$ (through $\mathbf{z}$).

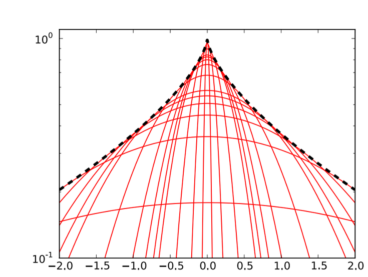


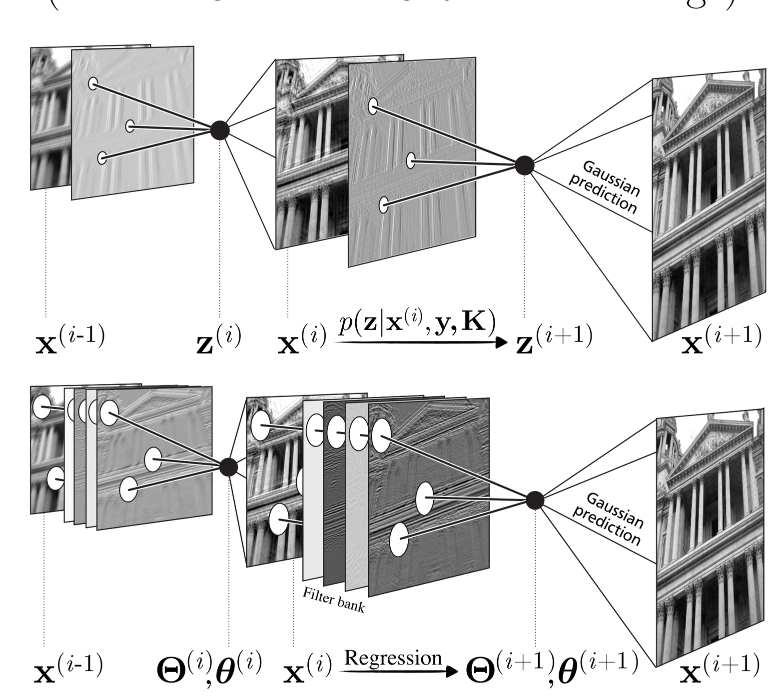
Figure 4: Half-quadratic representation of a sparse image prior.

Approach: Replace restricted half-quadratic inference with flexible discriminative prediction cascade (trained *Gaussian CRF* at each stage).

Figure 5: **Half-quadratic vs. discriminative cascade.**

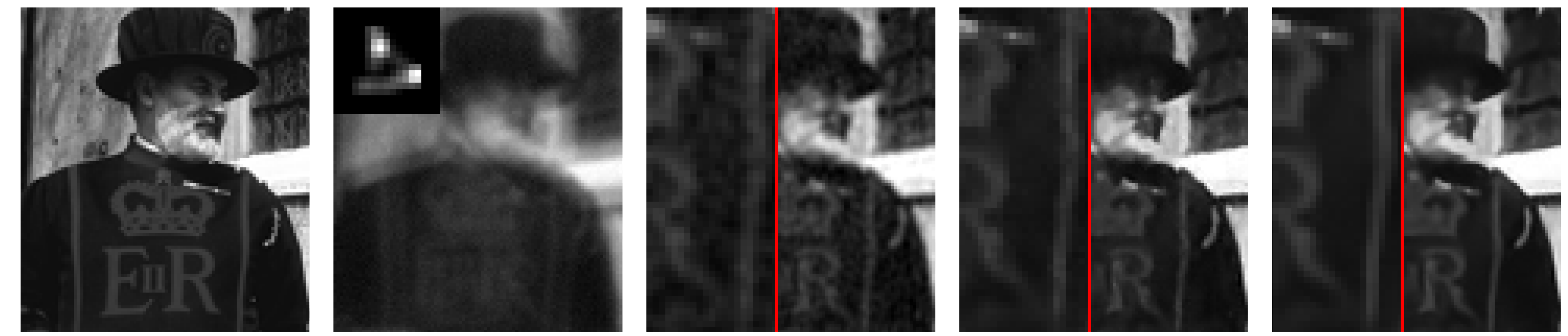
In half-quadratic deblurring *(top)*, $\mathbf{z}$ can only be updated based on pixels in the local clique of the MRF (small white circles).

In the proposed discriminative cascade *(bottom)*, arbitrary features over larger areas (large white circles) can be used to regress parameters $\boldsymbol{\Theta}^{(i)}$ and $\boldsymbol{\theta}^{(i)}$. Expect better results in fewer iterations due to increased flexibility.



- Previous Gaussian CRFs [3, 11] → one stage of proposed cascade
  - sufficient for simpler tasks (e.g. image denoising)
  - would likely benefit from iterative refinement

Deblurring example at different model stages



Ground truth | Blurred, 15.68dB | RTF$_1$, 25.39dB | RTF$_2$, 27.71dB | RTF$_6$, 28.20dB

Experimental Results

State-of-the-art performance at faster runtime

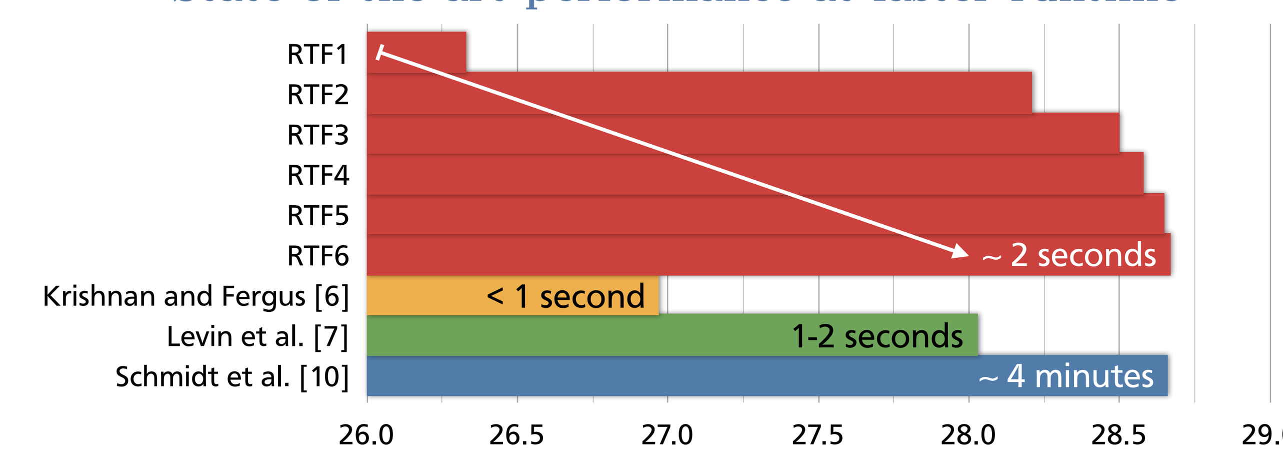


Figure 6: Average PSNR (dB) on 64 images from [10] (perfect blur kernels, $\sigma = 2.55$).

Superior results with estimated kernels

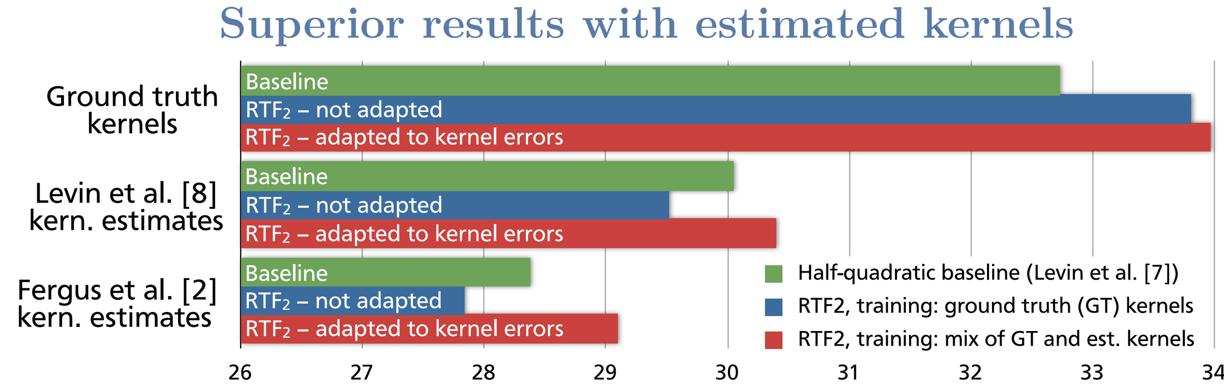


Figure 7: Average PSNR (dB) on 32 images from [8] (using estimated kernels).

Improvements for realistic higher-resolution images

	Image 1	Image 2	Image 3	Image 4
Kernel 01	+0.44	+0.54	+1.05	+0.76
Kernel 02	+0.44	+0.27	+0.38	+0.46
Kernel 03	+0.02	+0.03	+0.39	-0.26
Kernel 04	+0.31	+0.30	+0.61	+0.27
Kernel 05	+0.61	+0.44	+0.64	+0.05
Kernel 06	+0.40	+0.41	+1.03	+0.48
Kernel 07	+0.24	+0.55	+0.45	+0.31
Kernel 08	+0.76	+0.56	+2.17	+1.73
Kernel 09	+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

Figure 8: Performance gain (PSNR in dB) over results of Xu and Jia [12] in the benchmark of Köhler *et al.* [5]. Using kernel estimates of [12] with our non-blind approach, we can improve performance in 43 of 48 test instances, on average about 0.41dB.

Acknowledgements

We thank Pushmeet Kohli for discussions and Stephan Richter for help in preparation of Fig. 5. Parts of this work have been done when Uwe Schmidt interned at Microsoft Research Cambridge, UK. Funding is partly provided by Microsoft Research through its PhD Scholarship Programme.

Detailed Results

Method	σ = 2.55	σ = 7.65
Lucy-Richardson	25.38	21.85
Krishnan and Fergus [6]	26.97	24.91
Levin *et al.* [7]	28.03	25.36
5 × 5 FoE (MAP) [9]	28.44	25.66
Pairw. MRF (MMSE) [10]	28.24	25.63
3 × 3 FoE (MMSE) [10]	28.66	25.68

Stage	σ = 2.55	σ = 7.65
RTF$_1$	26.33	24.23
RTF$_2$	28.21	25.54
RTF$_3$	28.50	25.75
RTF$_4$	28.58	25.81
RTF$_5$	28.65	25.87
RTF$_6$	**28.67**	**25.89**

Table 1: *Training and testing with ground truth blur kernels.* Average PSNR (dB) on 64 images from [10] for two noise levels. Left half reproduced from [10].

Method	Kernels for training	Kernels for testing: GT	Levin [8]	Cho [1]	Fergus [2]
RTF$_1$	Ground truth (GT)	32.76	29.41	28.29	27.86
RTF$_2$	Ground truth (GT)	33.81	29.52	27.76	27.84
RTF$_1$	Mix of GT & Xu [12]	32.90	29.90	29.33	28.63
RTF$_2$	Mix of GT & Xu [12]	**33.97**	**30.40**	**29.73**	**29.10**
Levin [7]	—	32.73	30.05	29.71	28.38

Table 2: *Adaptation to kernel estimation errors at test time.* Average PSNR (dB) on 32 images from [8]. The last row shows baseline performance using [7].

Qualitative Example

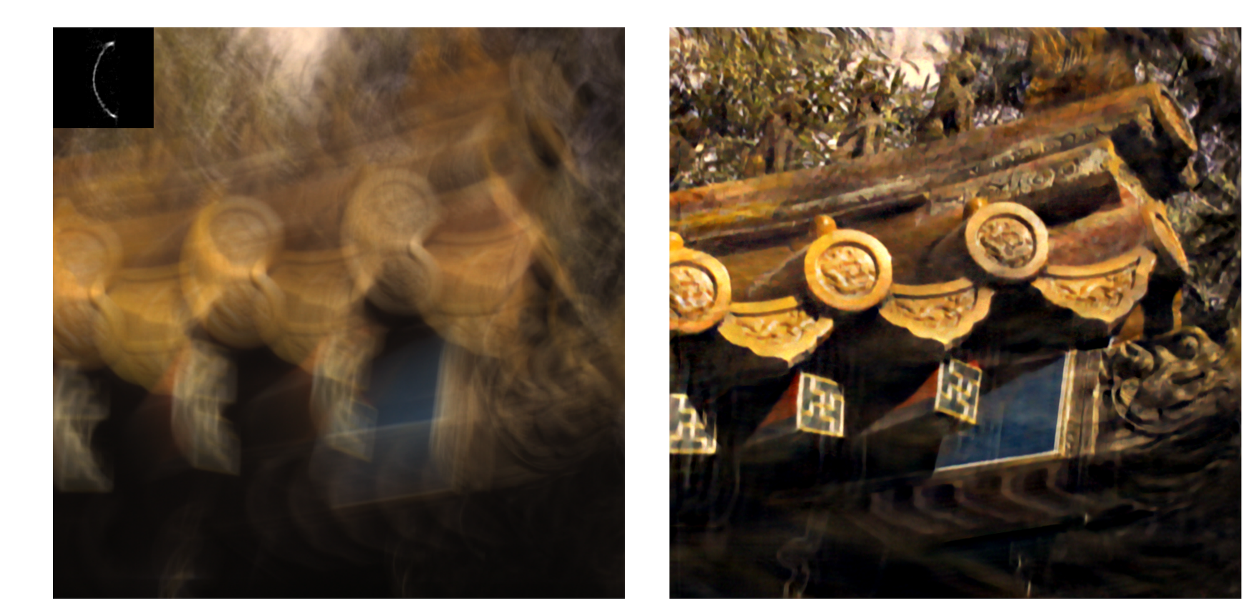


Figure 9: *Example for realistic higher-resolution image from [5]*, showing the result of our RTF$_2$ model *(right)* given blurred image *(left)* and kernel estimate by [12] *(top left)*.

References

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

[2] 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.

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

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

[5] 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.*

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

[7] 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.

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

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

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

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

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